Sistema de pesagem para indivíduos com mobilidade diminuída

Tiago Faustino Andrade

Relatório do Projecto Final

Orientadores:

Prof. Manuel Rodrigues Quintas

Inv. Carlos Manuel Sousa Moreira da Silva

Faculdade de Engenharia da Universidade do Porto

Mestrado Integrado em Engenharia Mecânica

Julho de 2008

Aos meus pais, avó, tio e à honestidade.

Resumo

A necessidade de prescrição de medicamentos manipulados, ou seja, feitos por farmacêuticos e adequados a cada paciente, e o simultâneo acompanhamento da evolução da massa corpórea de doentes com mobilidade diminuída, acamados por longos períodos, obriga à sua pesagem *in situ*.

O objectivo deste trabalho consiste no desenvolvimento de um sistema de pesagem, para indivíduos com mobilidade diminuída, portátil, manuseabilidade simples, baixo custo e com uma capacidade até 5000 N.

Através da *internet* pesquisaram-se dispositivos para pesagem da massa corpórea de indivíduos, existentes no mercado, comparando-se as suas características. Estudaram-se algumas normas aplicadas a sistemas de pesagem e a ambientes hospitalares, tendo em vista a acreditação do produto a desenvolver neste trabalho.

A escolha do elemento sensor recaiu sobre um do tipo piezoresistivo da empresa *Tekscan's*: o *FlexiForce A201*, devido ao seu simples condicionamento de sinal e ao seu baixo custo.

A elaboração de alguns cálculos preliminares com o intuito de objectivar uma ideia do projecto mecânico deu origem, após processo iterativo, à estrutura mecânica do dispositivo de pesagem, elaborada no *SolidWorks.* Esta foi evoluindo à medida que se iam afinando os componentes constituintes através do programa *COSMOSWorks*. Paralelamente foi desenvolvido o projecto eléctrico e electrónico, finalizado com a montagem dos componentes nas placas de circuito impresso desenhadas no programa *Eagle 5.0.0*. A placa de desenvolvimento *EasyPIC4* apoiou os primeiros testes ao circuito assim como permitiu a programação do microcontrolador. A integração da componente eléctrica, electrónica e informática deu origem a uma consola de comando de simples utilização, permitindo a interacção do operador com o dispositivo de pesagem projectado.

Development of a system to weight people with reduced mobility

Abstract

The need to prescribe custom made medication, and the simultaneous monitoring of reduced mobility patients' mass, confined in bed for long periods of time, leads to their in situ weight evaluation.

The objective of this work consists on the development of a weight measuring system, designed for individuals with reduced mobility capabilities, that is portable, has a simple interface, a low cost and a weighting capacity of up to 5000 N.

A research over the internet was firstly conducted to find similar weight measuring devices already existent in the market, in order to compare their characteristics. Then, a further study on standards applied to weight measurement in a hospital environment was held, so that the final product of this project is accepted by the community as a valid instrument.

As for the sensor used in the project, the Tekscan's Flexiforce A201 piezoresistive sensor was chosen, due to its simple signal conditioning capabilities and its low cost.

Preliminary calculations in order to define the mechanical project after an interactive process, lead to the device's mechanical structure, designed using SolidWorks. This structure evolved a process of fine tuning of their different components that was performed using COSMOSWorks. Simultaneously the electric and electronic project was developed, resulting on their assembly in printed circuit boards (PCB) designed using Eagle 5.0.0. An EasyPIC4 development board allowed the first testing of the circuit boards, as well as the microcontroller programming. The integration between the electrical, electronic and informatic components resulted in a command line prompt as a simple user interface, allowing interaction between the operator and the developed weight measuring device.

Agradecimentos

Quero agradecer aos meus orientadores, Professor Manuel Rodrigues Quintas e Investigador Carlos Moreira da Silva, pelas discussões proporcionadas nos momentos de maior dificuldade e por me mostrarem, neste realismo académico, o que é ser-se engenheiro.

Um especial obrigado ao Engenheiro Jorge Reis por me instruir, nunca receando transmitir o seu vasto conhecimento.

Aos meus colegas de laboratório agradeço a paciência e compreensão mantida, ao longo do semestre, na escolha da discografia ouvida.

Obrigado aos colegas que nunca me negaram os seus apontamentos e me ajudaram com as suas verdades, permitindo-me estar agora a escrever estas palavras.

Índice de conteúdos

Índice de figuras ..... *ix*

Índice de tabelas ..... *xiii*

Índice de Gráficos ..... *xv*

1 . Introdução Geral e Objectivos *1*

1.1 Motivação **1**

1.2 Objectivos **3**

1.3 Estrutura do relatório **3**

2 . Estado da Arte *5*

2.1 Introdução **5**

2.2 Balanças hospitalares existentes e suas características **5**

2.2.1 Scale-Tronix 2250CE 7

2.2.2 Seca 984 7

2.2.3 INTERCROMO 7727 8

2.2.4 SOEHNLE 7710/7711 9

2.2.5 Coventry Scale Company bed scale 10

2.2.6 Algen Scale SR193 10

2.3 Normas **11**

2.4 Conclusão **11**

3 . Princípios de Funcionamento *13*

3.1 Introdução **13**

3.2 Sensorização do elemento de pesagem **13**

3.3 Princípio de funcionamento da balança **19**

3.4 Conclusão **24**

4 . Projecto Mecânico *27*

4.1 Introdução **27**

4.2 Dimensionamento mecânico 27

4.3 Tesoura 31

4.4 Sistema de elevação 35

4.4.1 Alavanca telescópica 35

4.4.2 Sistema de retenção do veio escalonado 38

4.5 Sistema de alojamento do elemento sensor 41

4.6 Conclusão 44

5 . Projecto Eléctrico/Electrónico e Informático 45

5.1 Introdução 45

5.2 Circuito electrónico 45

5.2.1 Caracterização dos principais componentes 45

5.2.2 Esquemas de ligações 48

5.3 Programa desenvolvido 51

5.3.1 Sistema de desenvolvimento 51

5.3.2 Código desenvolvido 53

5.4 Consola de comando 57

5.5 Conclusão 58

6 . Conclusão Geral e Sugestões de Trabalhos Futuros 61

6.1 Conclusões 61

6.2 Sugestões para trabalhos futuros 62

7 . Referências Bibliográficas 63

7.1 Livros 63

7.2 Páginas da *internet* 63

Anexo A . Programa Desenvolvido 69

Anexo B . Análise Estática dos Componentes Mecânicos 99

Anexo C . Orçamento Eléctrico/Electrónico do Protótipo 103

Anexo D . Manual do Utilizador 105

Anexo E . Esquemas Eléctricos/Electrónicos 113

Índice de figuras

Figura 1. Tipos de balanças analisadas, Tipo 0, Tipo 1 e Tipo 2, respectivamente. 6

Figura 2. Sistema Scale-Tronix 2250CE. 7

Figura 3. Sistema Scale-Tronix 2250CE instalado. 7

Figura 4. Sistema de colocação do elemento de carga da Seca 984. 8

Figura 5. Sistema Seca 984. 8

Figura 6. Sistema INTERCROMO 7727. 9

Figura 7. Sistema SOEHNLE 7710. 9

Figura 8. Funcionamento da sapata SOEHNLE 7710. 9

Figura 9. Sistema da Coventry Scale Company. 10

Figura 10. Sistema de alojamento da célula de carga da Coventry Scale Company. 10

Figura 11. Sistema Algen Scale SR193. 10

Figura 12. Sensor piezoresistivo *FlexiForce*. 14

Figura 13. Circuito eléctrico do condicionamento de sinal do sensor preconizado pelo fabricante. 15

Figura 14. Ensaio ao elemento sensor piezoresistivo *FlexiForce*. 16

Figura 15. Dispositivo de pesagem. 19

Figura 16. Sistema de pesagem em vista explodida. 21

Figura 17. Dispositivo de pesagem, sapata, legendado. 23

Figura 18. Estudo do sistema de carga do sensor. 28

Figura 19. Estudo do dispositivo de pesagem. 28

Figura 20. Esquematização do processo de elevação. 28

Figura 21. Polígonos de forças. 28

Figura 22. Aplicação das forças horizontais no sistema. 30

Figura 23. Esquematização do sistema de alavanca. 31

Figura 24. Vista isométrica e explodida da estrutura mecânica, tesoura. 32

Figura 25. Distribuição das tensões na estrutura mecânica com um factor de escala de 100. 33

Figura 26. Distribuição das deslocações sofridas pela tesoura com um factor de escala de 100..... 34

Figura 27. Distribuição das tensões no veio do rolete com um factor de escala de 112..... 34

Figura 28. Vista isométrica e explodida da alavanca telescópica. 35

Figura 29. Distribuição das tensões na biela com um factor de escala de 1. 36

Figura 30. Distribuição das tensões na manivela durante a elevação da cama com um factor de escala de 1. 37

Figura 31. Ilustração do ponto morto..... 38

Figura 32. Vista isométrica e explodida do sistema de retenção do veio escalonado. 39

Figura 33. Distribuição das tensões na guilhotina e no veio escalonado com um factor de escala de 126. 40

Figura 34. Vista isométrica e explodida do sistema de alojamento do sensor, cassete. 41

Figura 35. Distribuição das tensões na cassete com um factor de escala de 800. 42

Figura 36. Distribuição das deslocações na cassete com um factor de escala de 1. 43

Figura 37. Distribuição das deslocações da pata da cassete com um factor de escala de 1.... 43

Figura 38. Pinos do microcontrolador. 46

Figura 39. Diagrama de blocos dos circuitos eléctricos/electrónicos. 47

Figura 40. Circuito eléctrico constituinte da consola de comando. 49

Figura 41. Circuito eléctrico constituinte de cada uma das sapatas..... 49

Figura 42. Placa de circuito impresso da consola. 50

Figura 43. Placa de circuito impresso de cada uma das tesouras..... 50

Figura 44. Aspecto do ambiente de trabalho do *MikroC*. 51

Figura 45. Compilador *PicFLASH*..... 52

Figura 46. Placa de desenvolvimento 52

Figura 47. Teclado utilizado na consola. 53

Figura 48. Diagrama de blocos simplificado do programa desenvolvidos. 53

Figura 49. Visualização de uma memória das memórias. 54

Figura 50. Diagrama de blocos simplificado do menu "Tarar" 55

Figura 51. Diagrama de blocos simplificado do menu pesar. 56

Figura 52. Vistas de cima, trás e frente da consola de comando. 57

Figura 53. Interior da consola de comando. 58

Figura 54. Elementos para programação do microcontrolador montado na consola de comando. 58

Figura 55. Distribuição das deslocações no veio do rolete com um factor de escala de 112. . 99

Figura 56. Distribuição das tensões no suporte do rolete e respectivo veio com um factor de escala de 1 99

Figura 57. Distribuição dos deslocamentos no suporte do rolete e respectivo veio com um factor de escala de 1 99

Figura 58. Distribuição dos deslocamentos na biela com um factor de escala de 1 100

Figura 59. Distribuição das tensões na pata da cassete com um factor de escala de 1 100

Figura 60. Deformações na tesoura com um factor de escala de 100. 100

Figura 61. Deformações no veio do rolete com um factor de escala de 100. 101

Figura 62. Deformações na biela com um factor de escala de 100. 101

Figura 63. Deformações na manivela com um factor de escala de 50. 101

Figura 64. Deformações no conjunto guilhotina/veio escalonado com um factor de escala de 100. 102

Figura 65. Deformações na cassete com um factor de escala de 100. 102

Figura 66. Deformações na pata da cassete com um factor de escala de 100. 102

Figura 67. Circuito eléctrico da consola. 113

Figura 68. Circuito eléctrico da sapata. 114

Índice de tabelas

Tabela 1. Cálculo do IMC. 2

Tabela 2. Riscos do excesso de peso. 2

Tabela 3. Quadro comparativo de balanças. 6

Tabela 4. Dados físicos do sensor piezoresistivo *FlexiForce* 14

Tabela 5. Características do sensor piezoresistivo *FlexiForce* 15

Tabela 6. Lista de peças 20

Tabela 7. Evolução das forças presentes no dispositivo de pesagem para diferentes tamanhos de rodas 29

Tabela 8. Características do microcontrolador 46

Tabela 9. Funções e valores associados a cada tecla. 56

Tabela 10. Orçamento eléctrico/electrónico do protótipo. 103

Índice de Gráficos

Gráfico 1. Curva de evolução da resistência em função da força aplicada. 15

Gráfico 2. Curva de evolução da condutância em função da força aplicada. 15

Gráfico 3. Primeiro ensaio realizado ao elemento sensor piezoresistivo *FlexiForce* 17

Gráfico 4. Segundo ensaio realizado ao elemento sensor piezoresistivo *FlexiForce* 17

Gráfico 5. Ensaio com incremento de cargas. 18

Gráfico 6. Ensaio com decremento de cargas. 18

Gráfico 7. Distância das rodas ao solo em função do seu raio. 24

Gráfico 8. Reacção horizontal em função dos raios das rodas. 30

1 . Introdução Geral e Objectivos

1.1 **Motivação**

A necessidade de prescrição de medicamentos, nomeadamente os manipulados, ou seja, medicamentos feitos por farmacêuticos à medida de cada paciente, e a simultânea monitorização da evolução da massa corpórea de indivíduos com mobilidade diminuída, acamados por longos períodos de tempo, obriga à sua pesagem *in situ*. Para tal recorre-se a balanças de simples manipulação e preferencialmente portáteis, desejavelmente instaláveis por um único operador e sem movimentação do doente.

Dos sistemas actualmente disponíveis, poucos são os que cumprem tais especificações, têm preços excessivamente elevados, o que consequentemente tem limitado a sua proliferação. No entanto, novos tipos de sensores, actualmente existentes no mercado, podem permitir soluções de custo substancialmente mais reduzido.

A garantia de uma boa pesagem com grau de precisão que ronda as 200 gf nos aparelhos existentes, facilita, com inúmeras vantagens, a prescrição de medicamentos, principalmente manipulados, a este tipo de indivíduos. Através destes medicamentos os tratamentos poderão ter maior eficácia, pois atendem às necessidades individuais de cada paciente. Permitem assim como permitem uma melhor compatibilidade, aquando da necessidade de tomar mais do que um tipo de medicamento e mesmo colmatar a falta deles no mercado. Economicamente, são mais rentáveis, pois a dosagem é mais precisa. Para além disso, são comercializados directamente com o cliente; no entanto o seu custo depende de vários factores, dos honorários da preparação, do valor das matérias-primas e dos valores dos materiais de embalagem. Este ramo comercial é seguro dado que os actuais padrões de qualidade na utilização dos mesmos se encontram enquadrados num sistema de boas práticas, a observar na preparação de medicamentos manipulados, de acordo com as normas e requisitos aprovados pelo INFARMED (http://www.infarmed.pt).

Com a pesagem do indivíduo, é possível obter o Índice de Massa Corporal (IMC) que indica se o indivíduo adulto está acima (obeso) ou abaixo do peso ideal e consequentemente considerado saudável ou não. Pode-se obter o IMC recorrendo-se à tabela 1, mostrada mais adiante, ou então calcular o seu valor utilizando-se a equação 1:

$$IMC = \frac{peso}{(altura)^2} \qquad \text{(eq. 1)}$$

Naturalmente que o IMC tem alguns pontos fracos, pois pessoas musculadas podem ter um IMC alto e não serem obesas. Este índice não é aplicável a crianças, e varia com as diferenças raciais e étnicas, como é referido no portal http://www.copacabanarunners.net. Na tabela 2, podemos ainda verificar os riscos inerentes ao excesso de peso através da correspondência de cores entre a tabela 1 e esta. Ambas as tabelas foram retiradas do portal da Associação de doentes Obesos e Ex-obesos de Portugal (http://www.adexo.pt).

Tabela 1. Cálculo do IMC.

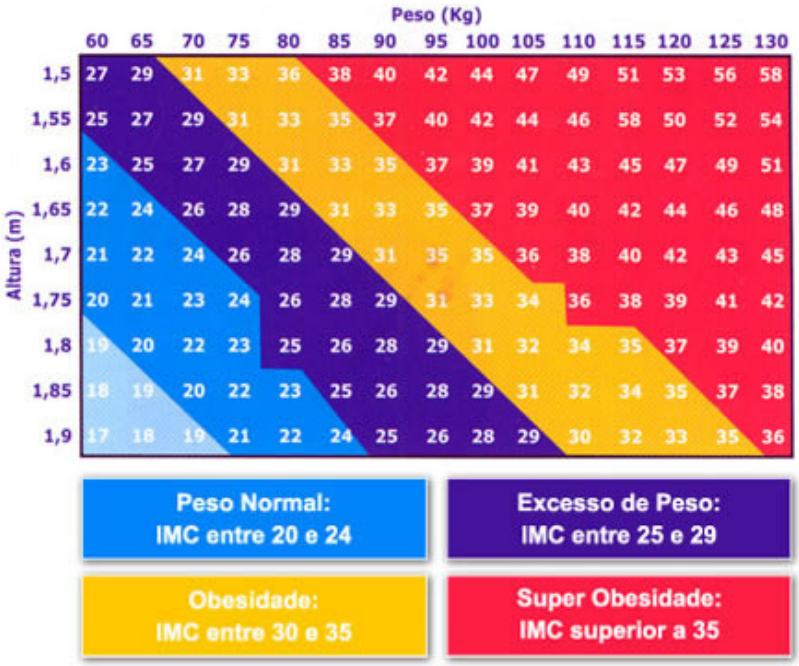


Tabela 2. Riscos do excesso de peso.

Super Obesidade: Risco Muito Aumentado	Obesidade: Risco Moderadamente Aumentado	Excesso de peso: Risco Ligeiramente Aumentado
Diabetes tipo 2	Doença coronária	Cancro
Colesterol elevado	Hipertensão	Síndroma do ovário poliquístico
Doença vesicular	Osteoartrose (joelho)	Alteração das hormonas da reprodução
Resistência à insulina	Hiperurícemia e gota	Diminuição da fertilidade
Apneia do sono		Dor lombar
		Malformações fetais associadas à obesidade materna

A necessidade de uma monitorização regular do paciente prende-se, essencialmente, com o acompanhamento da progressão de doenças, o que por vezes pode salvar vidas. Um caso comum é a perda de fluidos e necessidade de sua rápida reposição (http://www.scielo.br).

1.2 **Objectivos**

O objectivo deste trabalho é o de desenvolver um sistema de pesagem para indivíduos com mobilidade diminuída que responda aos seguintes requisitos:

- Portátil, de funcionamento de tipo electrónico com alimentação autónoma;
- Orientado para medição em camas hospitalares com rodas, eventualmente extensível a outros tipos de camas ou cadeiras com rodas;
- Capacidade de carga total até 5000 N, incluindo tara;
- Sistema centralizado de avaliação com leitura digital, permitindo compensação manual ou automática da tara;
- Manuseabilidade simples do sistema e sem recurso a mais do que um operador;
- Baixo custo de produção e reduzida necessidade de manutenção.

1.3 **Estrutura do relatório**

Para além desta parte introdutória, o presente documento está organizado em mais seis pontos, aos quais se seguem os anexos.

O ponto subsequente, denominado de *Estado da Arte*, pretende apresentar os diversos dispositivos de pesagem existentes no mercado, assim como o funcionamento dos diversos tipos de mecanismos utilizados, expondo as suas características principais de forma comparada.

Posteriormente, o ponto 3, denominado por *Princípios de Funcionamento*, debruça-se sobre a escolha e características do sensor utilizado no trabalho. Para além disso, apresenta a constituição mecânica da balança e a cinemática envolvida.

O *Projecto Mecânico*, pertencente ao quarto ponto, inicia com um pré dimensionamento que auxiliará nas opções tomadas durante a modelação 3D, no programa

SolidWorks, e validadas pelo programa *COSMOSWorks* onde se elabora um estudo estático tendo por base o Método de Elementos Finitos.

Seguidamente, o ponto 5, denominado de *Projecto Eléctrico/Electrónico e Informático*, incide sobre os componentes electrónicos utilizados e suas ligações, num primeiro ponto. Ulteriormente é descrito o código elaborado para programação do microcontrolador, finalizando-se este capítulo com a integração da electrónica e programação, dando origem a uma consola de comando.

No ponto 6 são retiradas conclusões, apresentando as limitações deste trabalho, e são dadas sugestões para a realização de trabalhos futuros.

Finalmente é apresentada, no ponto 7, a bibliografia utilizada na realização deste trabalho, à qual se seguem os anexos constituídos pelo código desenvolvido, estudos de simulação elaborados no programa *COSMOSWorks* de componentes mecânicos menos críticos e não apresentados no ponto quatro, orçamento eléctrico/electrónico do protótipo e manual do utilizador.

2 . Estado da Arte

2.1 Introdução

Com o intuito de realizar este trabalho iniciamo-lo pela sua contextualização e objectivação. Para isso é fundamental conhecer os dispositivos de pesagem da massa corpórea de um indivíduo, existentes no mercado, assim como as suas características, ou seja, o estudo do estado da arte. Deste modo foi elaborada uma pesquisa exaustiva, utilizando como ferramenta de trabalho motores de busca informáticos.

Este capítulo pretende apresentar os diversos dispositivos de pesagem existentes no mercado, os tipos de sistemas de funcionamento utilizados, expondo as suas características principais, proceder à sua comparação, alertar para a pesquisa e estudo das normas aplicáveis a este tipo de dispositivos de pesagem.

2.2 Balanças hospitalares existentes e suas características

A informação recolhida sobre algumas das balanças hospitalares existentes no mercado, destinadas a pesar indivíduos com mobilidade diminuída, forneceu algumas das suas características compiladas na tabela 3. Foram organizadas de forma a comparar alguns pontos relevantes entre balanças de iguais e diferentes géneros, tendo sido atribuída uma classificação, pelo autor, quanto aos tipos de balanças existentes. Do tipo 0, fazem parte as balanças que mais se assemelham ao dispositivo que se pretende desenvolver, ou seja, um sistema de pesagem que não obrigue o paciente a movimentar-se na sua cama e no espaço hospitalar. As do tipo 1, são balanças que obrigam a movimentação do corpo do paciente, ou seja, implica rolar o paciente de forma a colocá-lo sobre uma tela que será posteriormente içada. As do tipo 2, por seu lado, obrigam à deslocação da cama sobre a qual repousa o paciente, até uma balança colocada no chão. Para exemplo desses três tipos de balanças é apresentada uma figura 1, mais adiante. É de salientar que a coluna quatro, denominada de *Resolução*, contém os valores que foram designados de diferentes formas pelos fabricantes, como sendo, exactidão, graduação, resolução, precisão e divisões. Deste modo, não haverá uniformidade do termo, utilizado para denominar a coluna, quando forem descritas as características de cada uma das balanças do tipo 0.

Reserva-se a descrição de detalhe às balanças deste tipo, pois são essas as mais importantes para este trabalho, isto porque se assemelham mais ao sistema que se pretende desenvolverem.

Tabela 3. Quadro comparativo de balanças.

Tipo	Marca	Modelo	Resolução [kgf]	Peso máx. [kgf]	Preço
0	Scale-Tronix	2250CE	0,1	600	4.044,00 €
0	Seca	984	0,2	500	5.643,00 €
0	INTERCROMO	7727	0,1	500	3.850,00 €
0	SOEHNLE	7710/7711	0,2	500	6.597,36 €
0	Coventry Scale Company	bed scale	0,2	600	2.964,00 €
0	Algen Scale	SR 193	0,1	544	2.218,00 €
1	Scale-Tronix	2002	0,1	250	n.c.
1	DETECTO	IB600	0,1	272	3.762,00 €
2	Scale-Tronix	6202D	0,2	400	n.c.
2	Seca	656	0,1	360	2.535,00 €

Na figura 1, apresentam-se os três tipos de equipamentos existentes no mercado de forma a melhor se entender o texto anterior.

Figura 1. Tipos de balanças analisadas, Tipo 0, Tipo 1 e Tipo 2, respectivamente.

Nos pontos que se seguem são apresentadas as características das balanças do tipo 0, assim como figuras e comentários comparativos dos sistemas.

2.2.1 Scale-Tronix 2250CE

A balança da Scale-Tronix foi desenvolvida para uma monitorização contínua do peso, uma característica bastante importante que não é referida directamente nas outras. A colocação dos elementos de pesagem por baixo das rodas não é das mais triviais, pois há necessidade de deslocar a cama para cima dos mesmos, o que pode provocar escorregamentos das sapatas. Cada sapata, onde está inserido o sensor de carga, pesa 2,5 kgf e mede 121 mm x 291 mm. A tara é feita através de um botão que leva o valor do peso a zero, não havendo referências à possibilidade de memorização de diferentes taras.

A exactidão desta balança é de 100 gf para um peso máximo de 600 kgf, isto quer dizer que o paciente e a cama, com os equipamentos associados, não deverão pesar mais do que os 600 kgf.

A leitura do peso é feita em quilogramas força, através de um mostrador digital de LEDs, podendo ser ligado a um computador através de uma porta RS-232, considerada um extra.

Nas figuras 2 e 3 podemos ver a aparência deste dispositivo.

Figura 2. Sistema Scale-Tronix 2250CE.

Figura 3. Sistema Scale-Tronix 2250CE instalado.

2.2.2 Seca 984

Esta balança é constituída por quatro sapatas, tendo cada uma delas uma célula de carga, colocadas por baixo das rodas através de um sistema mecânico simples, e supostamente sem esforço por parte do operador. Utiliza baterias recarregáveis que, no caso de falta de energia na rede, mantém os valores previamente determinados. Possui uma função que

notifica, em tempo útil, o pessoal hospitalar no momento em que ocorram variações anormais da massa corporal.

A balança tem uma capacidade de 500 kgf. Possui uma graduação de 100 gf para um carregamento até 200 kgf e de 200 gf para uma carga compreendida entre 200 kgf e 500 kgf, sendo a gama da tara de 0 a 300 kgf.

O dispositivo onde está contida a célula de carga têm uma dimensão de 152 mm x 381 mm x 203 mm.

As figuras 4 e 5 que a seguir se apresentam ilustram este dispositivo.

Figura 4. Sistema de colocação do elemento de carga da Seca 984.

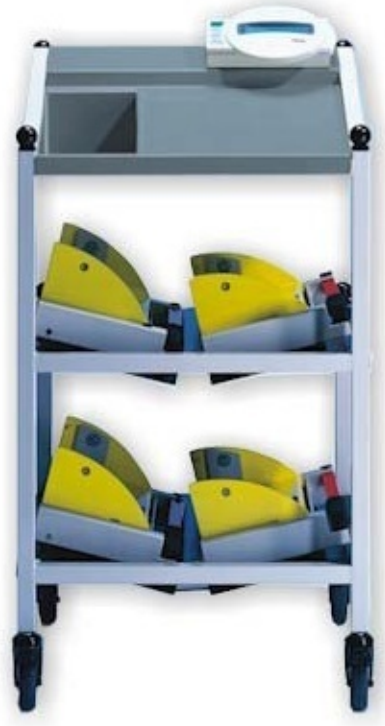


Figura 5. Sistema Seca 984.

2.2.3 INTERCROMO 7727

A colocação dos elementos de pesagem nesta balança é igual ao sistema da Scale-Tronix: deslocação da cama para cima do sistema que aloja os elementos sensores. Possui uma memória para 10 taras, limitadas a uma carga máxima de 300 kgf. Tem uma capacidade máxima de 500 kgf, à qual corresponde uma precisão de 100 gf.

A figura 6 apresenta o sistema descrito anteriormente, aplicado a uma cama hospitalar.

Figura 6. Sistema INTERCROMO 7727.

2.2.4 SOEHNLE 7710/7711

Com características bastante semelhantes às balanças anteriores, no que diz respeito a cargas máximas e resolução (divisões, segundo a SOEHNLE), o fabricante refere que esta, foi aprovada para EU Classe III, MPG Classe 1. Possui um mecanismo interessante de elevação que consiste numas sapatas que contêm de um lado uma rampa e do outro um rolete associado a uma alavanca, permitindo empurrar a roda ao longa da rampa, conseguindo assim afasta-la do solo. Contudo possui uma grande limitação: foi construída para levantar rodas de apenas 150 mm de diâmetro. Apresentada sobre a forma de dois modelos, o 7710 difere do 7711 no tipo de comunicação com a consola central. O modelo 7710 comunica através de cabos, enquanto o outro comunica por *Bluetooth*.

Na figura 7 é apresentado o sistema de pesagem aplicado a uma cama e na figura 8 são apresentados os diversos passos para a colocação das sapatas e elevação da roda.

Figura 7. Sistema SOEHNLE 7710.

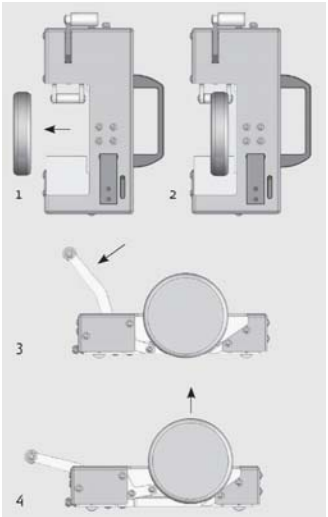


Figura 8. Funcionamento da sapata SOEHNLE 7710.

2.2.5 Coventry Scale Company bed scale

Constituída, mais uma vez, por quatro sapatas, colocadas da mesma forma que a INTERCROMA e a Scale-Tronix. Possui dimensões suficientes para acomodar as camas com rodas mais largas. É de fácil instalação e tem uma capacidade máxima de 600 kgf com uma resolução de 200 gf, estando aprovada na Classe III EU.

Na figura 9, podemos ver o sistema da INTERCROMO armazenado num veículo de apoio e na figura 10 uma das sapatas onde se aloja a célula de carga.

Figura 9. Sistema da Coventry Scale Company.

Figura 10. Sistema de alojamento da célula de carga da Coventry Scale Company.

2.2.6 Algen Scale SR193

Possuidora de um sistema extensível, desde 460 mm a 810 mm, na versão padrão, e de 810 mm a 1220 mm na versão superior, acomoda rodas com uma gama de diâmetros de 50,8 mm até 254 mm. Usufrui de quatro transdutores que medem, com uma exactidão de 100 gf, um peso máximo de 544 kgf, sendo este valor a combinação do peso da cama e do paciente.

A figura 11 mostra a consola e os dois sistemas extensíveis, que alojam os transdutores de pressão.

Figura 11. Sistema Algen Scale SR193.

2.3 **Normas**

Um dos pontos a ter em conta são as normas que regem este tipo de aparelhos. É importante fazer uma pesquisa exaustiva de normas existentes de forma a enquadrá-las com o equipamento que se pretende desenvolver para que seja reconhecido formalmente por um organismo de acreditação.

Um tipo de norma a ter em conta prende-se com a precisão das balanças, podendo-se enquadrar em quatro classes distintas como ser visto no Nº78-1-4-1996 do Diário da República-1 Série-B (http://www.amla.pt/metrologia.html). As classes são atribuídas em função do número de divisões, ou seja, do valor que se encontra quando dividimos a carga máxima da balança pela sua menor divisão, expressas na mesma unidade. Como forma de se entender melhor a atribuição da classe a uma balança, considere-se como exemplo, uma balança com capacidade de carga máxima de 500 kgf e com menor divisão de 200 gf (0,2 kgf). O número de divisões será 500 kgf a dividir por 0,2 kgf, perfazendo um total de 2500 divisões, valor com que se consultará uma tabela para atribuição da respectiva classe. A tabela não será mostrada, porque o Decreto-Lei consultado é uma actualização de um anterior, e este omite os títulos das colunas.

Outro tipo de normas a ter em conta prende-se com a compatibilidade electromagnética (EMC) dos equipamentos, que em ambientes hospitalares pode ser bastante crítico, uma vez que este sistema pode interferir com aparelhos de suporte da vida. Os ensaios de EMC podem ser realizados no Laboratório de Compatibilidade Electromagnética (LCEM) da ANACOM, acreditado pelo IPAC, Instituto Português de Acreditação.

2.4 **Conclusão**

Nos diversos dispositivos aqui apresentados há dois que se destacam na vertente do projecto mecânico, a balança da Seca e a da SOEHNLE. Observou-se que é comum a cama ser empurrada para cima do sistema de pesagem. Embora seja uma solução que facilita o projecto mecânico, dificulta a sua instalação, pois as sapatas não estão rigidamente fixas ao chão. No entanto é possível que tenham na sua base um revestimento antideslizante. A vantagem observada na solução da Seca e da SOEHNLE é a não necessidade de deslocamento da cama no plano horizontal, isto porque muitas vezes há equipamentos e cabos associados que dificultam a sua movimentação, por mais pequena que seja, exigindo também menor esforço por parte do operador hospitalar.

Todos os sistemas apresentam uma interface amigável e com funcionalidades muito parecidas. Há que destacar a função do sinal sonoro emitido quando existem variações de massa corpórea, nos modelos da Algen, Scale-Tronix e Seca, podendo existir também nas outras, no entanto a informação não é disponibilizada.

Outra função importante é a memorização de mais do que uma tara e neste caso só a INTERCROMO é que disponibiliza essa informação, como foi apresentado anteriormente.

Ter um sistema que possua baterias é importante, pois pode dar-se o caso de ocorrer uma falha eléctrica. Existindo esta funcionalidade poderá existir continuidade na monitorização da massa corpórea, assim como a manutenção dos dados guardados anteriormente. Possibilita por outro lado a libertação de tomadas que possam ser necessárias para outros dispositivos. Esta funcionalidade é omissa em todos os casos à excepção da marca Seca, que possui uma bateria para manter os valores guardados.

A existência de um sistema "sem fios" constitui uma solução interessante, disponibilizada pela SOHENLE, permitindo assim evitar cabos espalhados pelo chão. Contudo, deve-se ter cuidado com as possíveis interferências electromagnéticas que o sistema "sem fios" pode introduzir no ambiente hospitalar.

De uma forma geral todas as balanças apresentam características muito idênticas. A optar por uma delas deverá ter-se em conta, acima de tudo, a facilidade de manuseabilidade, o preço, peso máximo e resolução.

A pesquisa feita por normas, não foi suficientemente exaustiva, contudo permitiu ter a noção das implicações a que pode estar sujeito este sistema, relativamente à compatibilidade electromagnética e às classificações a que está sujeito, caso se pretenda homologar o equipamento.

3 . Princípios de Funcionamento

3.1 Introdução

Este capítulo incide sobre as principais características do elemento sensor e o funcionamento geral do sistema mecânico projectado neste trabalho.

A escolha e consequente utilização do tipo de sensor utilizado deveu-se à intenção de conhecer melhor o seu comportamento experimentalmente, pois aparenta ser uma solução promissora devido à sua flexibilidade. Para além disso está em curso um estudo e consequente desenvolvimento de um elemento sensor piezoresistivo envolvendo elementos da Secção de Automação, Instrumentação e Controlo (SAIC). Se a opção tomada foi ou não boa para esta aplicação, ficará para constatar mais adiante. Contudo, mesmo que se tenha de substituir este elemento sensor por outro ou por uma célula de carga, toda a estrutura mecânica manter-se-á, a menos do conjunto, "cassete", que aloja o elemento sensor.

O funcionamento mecânico aproxima-se do das balanças que possuem um sistema de elevação, isto é, apenas se coloca cada uma das sapatas junto das rodas actuando-se de seguida uma alavanca que as erguerá individualmente. A vantagem do sistema aqui desenvolvido consiste na versatilidade, admitindo uma gama de diâmetros de rodas compreendidos entre 100 mm e 200 mm, bem como a deslocação mínima do doente, uma vez que é somente necessário elevar a roda da cama alguns milímetros acima do chão.

3.2 Sensorização do elemento de pesagem

O elemento sensor utilizado neste trabalho é do tipo piezoresistivo, com uma capacidade de carga de 1000 lbf, aproximadamente 450 kgf, o *Tekscan's FlexiForce A201*. Como refere o fabricante, a forma de medição é por variação da resistência, 20 MΩ em vazio e 5 kΩ sob carga máxima, relativamente a uma força aplicada na sua área sensível, um círculo com 9,5 mm de diâmetro. Na sua construção utiliza duas camadas compostas por película de poliéster. Cada uma destas camadas contém um filme de prata condutora que delimita a área sensível, estendendo-se até ao conector de ligação do sensor ao exterior, constituído por tês pinos, tendo o pino do meio a função de descarga electrostática das películas. A área sensível,

circunscrita pela prata, é constituída por uma tinta sensível à pressão. Testes feitos pelo fabricante, com uma carga de aproximadamente 25 kgf e alojamento do sensor entre duas peças de metal, sugerem uma longevidade de um milhão de ciclos. Embora o fabricante não aconselhe o corte da fita a jusante da parte sensível, eles reportam casos de clientes que o fizeram com sucesso e sem qualquer problema, como também foi possível constatar à posteriori.

Na figura 12 é apresentado o elemento sensor acima descrito.

Figura 12. Sensor piezoresistivo *FlexiForce*.

Seguidamente são mostrados os dados deste elemento sensor, fornecidos pelo fabricante, organizados em duas tabelas: uma com os dados físicos (Tabela 4) e outra com as características (Tabela 5). Estas são seguidas de um circuito eléctrico (Figura 13) que contém o condicionamento de sinal preconizado pelo fabricante. Este conjunto de dados termina com dois gráficos, o primeiro com a curva de evolução da resistência em função da força aplicada (Gráfico 1) e o segundo com a curva da condutância em função da força aplicada (Gráfico 2).

Tabela 4. Dados físicos do sensor piezoresistivo *FlexiForce*.

Dados Físicos	
Comprimento	203 mm
Largura	14 mm
Diâmetro da área sensível	9,5 mm
Conector	3 pinos
Espessura	0,2 mm

Tabela 5. Características do sensor piezoresistivo *FlexiForce*.

Características	
Erro de linearidade	< ±5%
Repetibilidade	< ±2.5% (80% de força aplicada)
Histerese	< 4.5% (80% de força aplicada)
Desvio	< 5% em escala de tempo logaritmica (carga de 90%)
Tempo de resposta	< 5 microsegundos
Temperaturas de operacionalidade	-9°C - 60°C
Força	440 N (varia com o tipo de condicionamento de sinal)
Sensibilidade à temperatura	0.36% por °C)

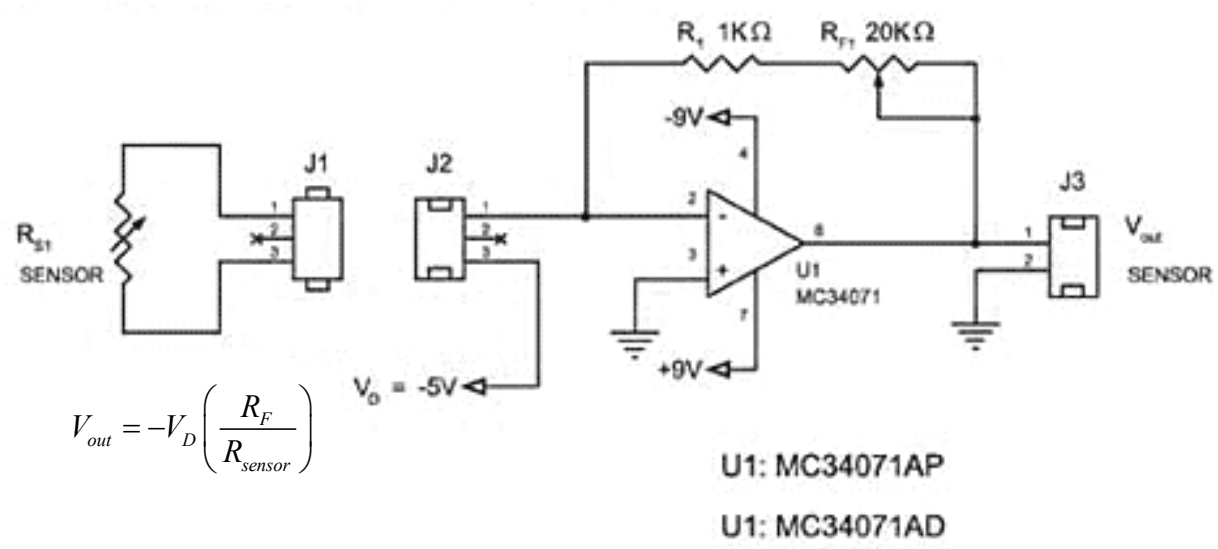


Figura 13. Circuito eléctrico do condicionamento de sinal do sensor preconizado pelo fabricante.

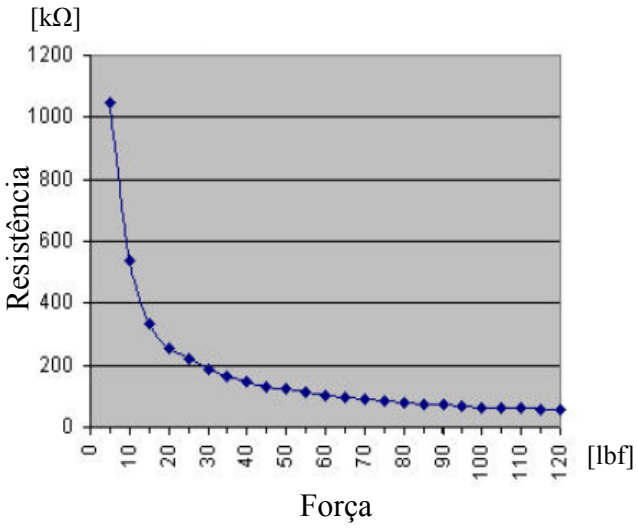


Gráfico 1. Curva de evolução da resistência em função da força aplicada.

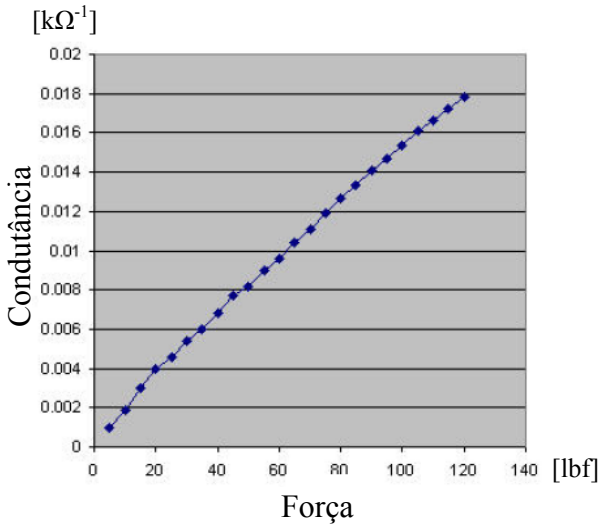


Gráfico 2. Curva de evolução da condutância em função da força aplicada.

Na tentativa de melhor compreender o comportamento do sensor, no que toca à sua histerese e repetibilidade, fez-se uma pequena análise experimental do seu comportamento. Mediu-se a resistência directamente entre pinos, através de um multímetro e sem qualquer tipo de amplificação do sinal, e aplicaram-se sucessivas cargas desde os 20 kgf até aos 100 kgf, medidas através de uma balança com resolução de 1 kgf, como se pode ver na figura 14.

Figura 14. Ensaio ao elemento sensor piezoresistivo *FlexiForce*.

Na primeira fotografia (a) referente à figura anterior, pode-se verificar que o elemento sensor (2) está colocado sobre uma superfície lisa (1). O carregamento é transmitido à superfície sensível do sensor através de uma pastilha polida em aço (3), que por sua vez é carregada pelo perfil em alumínio (4) que suporta as diferentes cargas. É importante o sensor estar compreendido entre superfícies lisas de forma a evitar a precoce degradação do mesmo. A fotografia (b) é uma ampliação da fotografia (c) e permite ver com maior detalhe o multímetro, a balança e as cargas utilizadas nos ensaios.

Nos dois gráficos (3 e 4) que se seguem mostram-se os resultados obtidos nos dois ensaios realizados.

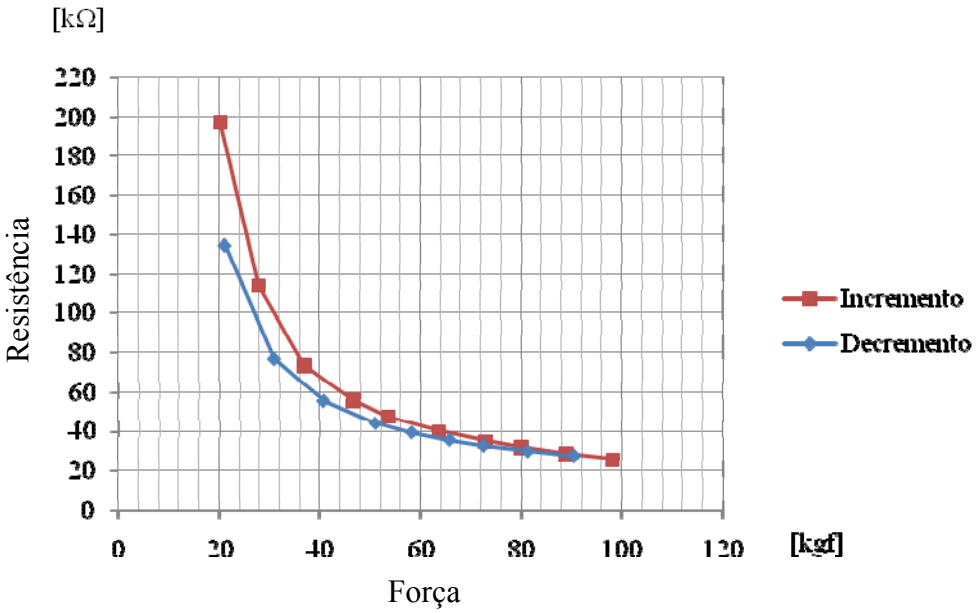


Gráfico 3. Primeiro ensaio realizado ao elemento sensor piezoresistivo *FlexiForce*.

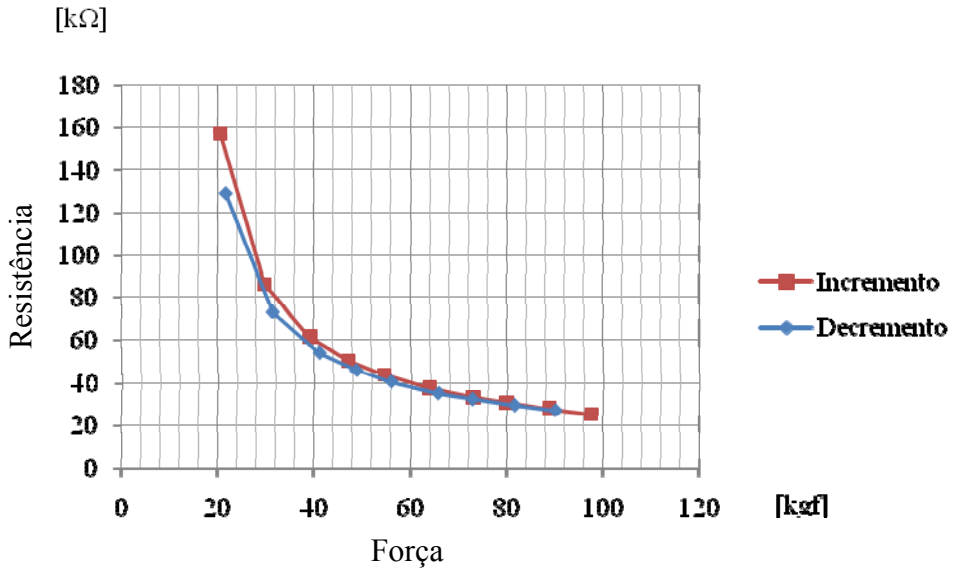


Gráfico 4. Segundo ensaio realizado ao elemento sensor piezoresistivo *FlexiForce*.

Embora sem rigor, devido ao tipo de balança utilizada, podem-se tirar algumas conclusões relevantes sobre o comportamento deste sensor, como se pode verificar nestes gráficos e nos seguintes. Nos gráficos 3 e 4 é possível observar a existência de histerese em

ambos os ensaios para cargas de valor mais baixo. É de notar que no primeiro ensaio a histerese está mais evidenciada, o que quer dizer que o comportamento do sensor se alterou de um ensaio para outro. Este facto poderá ser justificado pela estreia do sensor, ou seja, o sensor poderá necessitar de um determinado número de ciclos de carga e descarga para equilíbrio dos seus constituintes, traduzindo-se numa menor histerese e melhor repetibilidade. Convém referir que, embora de forma grosseira, foi feito, previamente, um teste de histerese à balança. Esta foi carregada com massas conhecidas e, posteriormente, descarregada, não tendo sido observada uma histerese significativa.

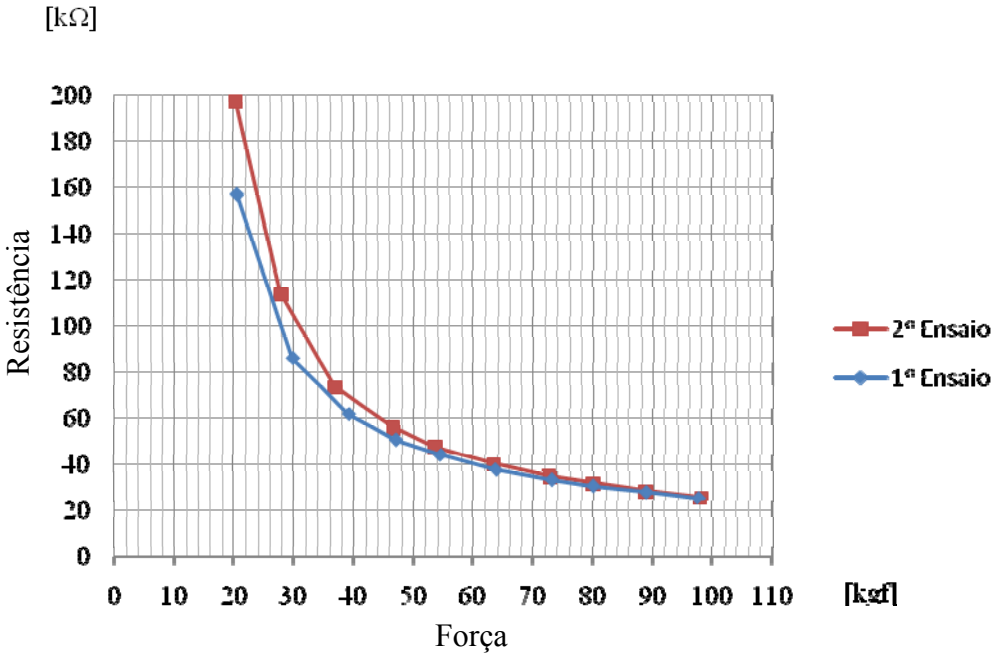


Gráfico 5. Ensaio com incremento de cargas.

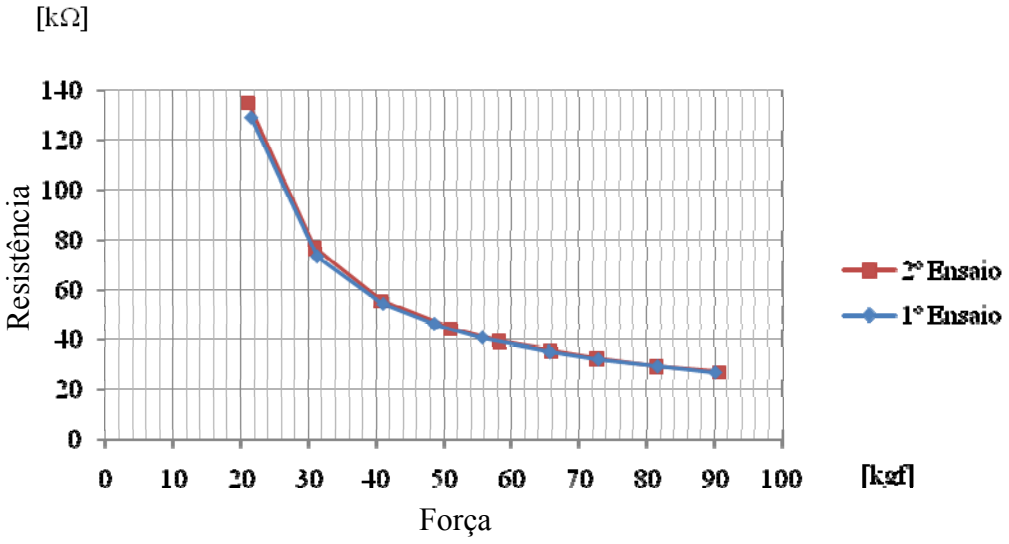


Gráfico 6. Ensaio com decremento de cargas.

No gráfico 5 observa-se um desfasamento entre os dois ensaios para cargas inferiores a 60 kgf, já no gráfico 6 pode-se considerar que as duas curvas são praticamente coincidentes ao longo do gráfico. Apesar de dois ensaios não serem suficientes para caracterizar o sensor é possível concluir que este tipo de sensor funciona melhor quando sujeito a uma pré-carga, antes de efectuar qualquer medição, como sugerido pelo fabricante.

O trabalho realizado foi idealizado para utilizar três sensores por sapata, o que resulta num total de doze sensores. Esta opção foi tomada com o intuito de reduzir a complexidade mecânica e consequente diminuição de custos, visto que cada sensor custa aproximadamente 20€, na compra de um lote de oito.

3.3 **Princípio de funcionamento da balança**

Os componentes que constituem esta balança, modelados em *SolidWorks*, estão representados na figura 15. Consultado o programa de modelação, obteve-se um atravancamento para a abertura máxima de 330 mm x 325 mm, largura e comprimento respectivamente. Mais adiante, apresenta-se uma vista explodida da balança, figura 16, acompanhada por uma tabela 6 que contém a lista de peças para uma das quatro sapatas. Posteriormente, serão apresentados em detalhe os diferentes sistemas que constituem este mecanismo.

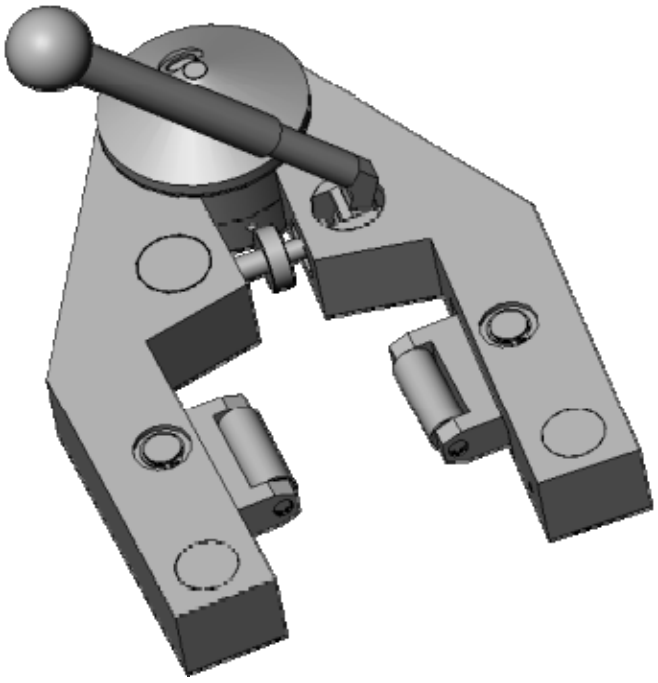


Figura 15. Dispositivo de pesagem.

Tabela 6. Lista de peças

Nº	DESIGNAÇÃO	NORMA ou DESENHO Nº	MATERIAL	Nº de Ref.
1	Botão de actuação		Al - 6061	1,1
2	Mola de compressão		50 CrV 4	1,2
1	Guilhotina		X 42 Cr 13	1,3
1	Corpo		X 5 CrNiMo 17 12 2	1,4
1	Veio escalonado		X 42 Cr 13	1,5
1	Volante			1,6
1	Base		X 5 CrNiMo 17 12 2	1,7
1	Tampa roscada		Al - 6061 - T6	2,1
1	Anilha de Aço			2,2
1	Braço superior		Al - 6061 - T6	2,3
1	Anilha INAFAG PAW62			2,4
1	Braço inferior		Al - 6061 - T6	2,5
2	Veio Φ18			2,6
2	Suporte do rolete		Al - 6061 - T6	2,7
1	Apoio roscado		X 5 CrNiMo 17 12 2	2,8
2	Casquilho INAFAG PAP18			2,9
2	Anilha INAFAG PAW18			2,10
2	Veio do rolete			2,11
6	Casquilho INAFAG PAP10			2,12
1	Tubo do rolete		X 5 CrNiMo 17 12 2	2,13
2	Tampa do braço		X 5 CrNiMo 17 12 2	2,14
1	Esfera roscada			3,1
2	Pino - 2x14	ISO 2338		3,2
1	Tubo extensível 1		X 5 CrNiMo 17 12 2	3,3
1	Tubo extensível 2		X 5 CrNiMo 17 12 2	3,4
1	Mola de tracção		50 CrV 4	3,5
1	Manivela		X 42 Cr 13	3,6
1	Veio de união roscado		X 5 CrNiMo 17 12 2	3,7
1	Cavilha Φ10			3,8
2	Cavilha Φ6			3,9
2	Biela		X 42 Cr 13	3,10
1	Base		X 5 CrNiMo 17 12 2	3,11
3	Apoio superior		X 5 CrNiMo 17 12 2	4,1
3	Lâmina rectificada		X 5 CrNiMo 17 12 2	4,2
3	Pino paralelo - 1x10	ISO 8734		4,3
3	Corpo da cassete		X 5 CrNiMo 17 12 2	4,4
3	Apoio inferior		X 5 CrNiMo 17 12 2	4,5
3	Esfera			4,6
3	Pata		X 5 CrNiMo 17 12 2	4,7
1	Pino roscado DIN 914 - M3x20	DIN 914		5,1
4	Anel elástico DIN 471 - 18x1,2	DIN 471		5,2
2	Anel elástico DIN 472 - 45x1,75	DIN 472		5,3
4	Anel elástico DIN 472 - 10x1	DIN 472		5,4
3	Pino roscado DIN 915 - M4x12	DIN 915		5,5
6	Parafuso ISO 7046 - M2x3	ISO 7046		5,6
3	Anel elástico DIN 472 - 36x1,5	DIN 472		5,7

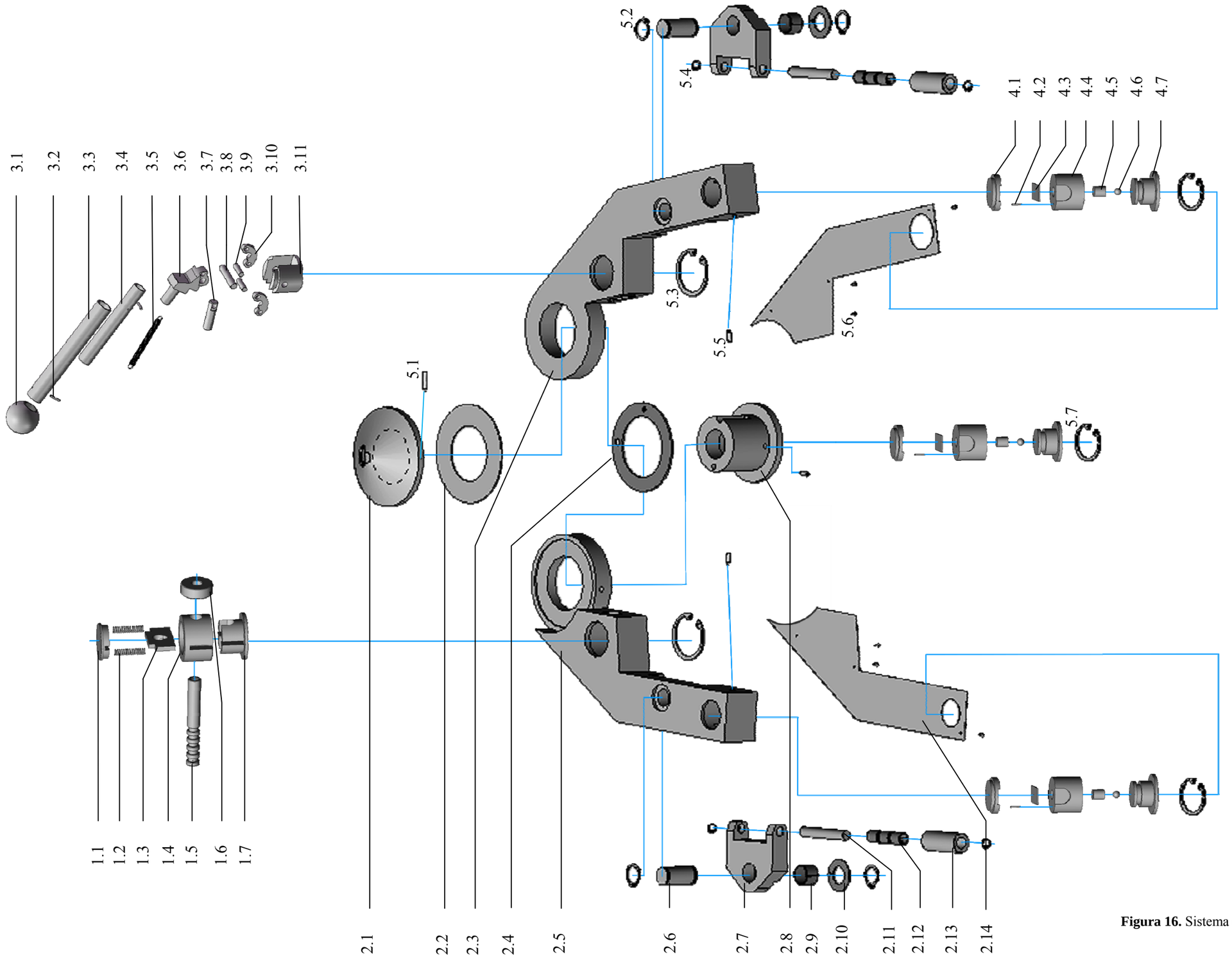


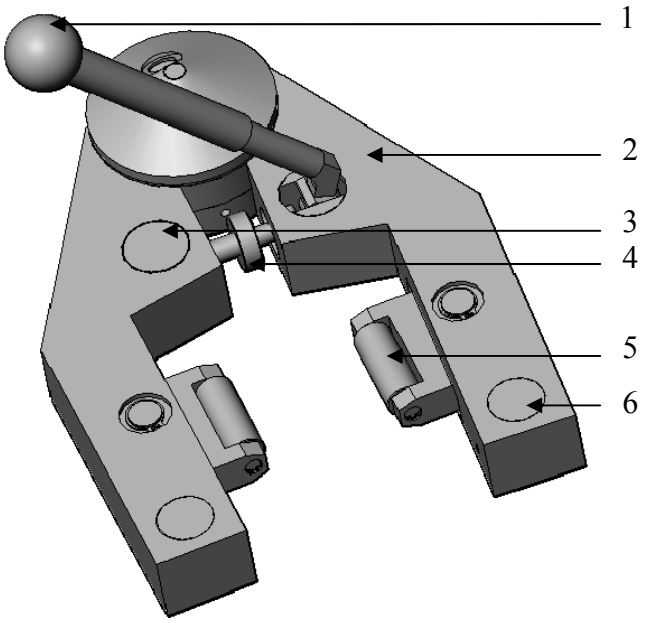
Figura 16. Sistema de pesagem em vista explodida.

O sistema é constituído por quatro sapatas, sendo cada uma delas carregada por cada uma das quatro rodas constituintes da cama hospitalar.

Na figura 17 estão representadas as partes "activas" da sapata com o intuito de facilitar a compreensão do modo de elevação da cama através deste sistema.

Cada sapata é constituída por uma estrutura principal, denominada por tesoura, onde se articulam dois roletes de suporte da roda que graças à sua rotação, se mantêm, paralelos entre si.

Uma parcela do peso da cama é sentido por três sensores, colocadas em triângulo, um no ponto de articulação da tesoura e os outros dois na extremidade dos braços da mesma. Estes sensores estão alojados em suportes idênticos que designamos por "cassete", e que integram os três pontos de apoio de cada sapata, para que toda a carga aplicada ao sistema passe pelos sensores.



Legenda:

1. Alavanca
2. Tesoura
3. Botão
4. Volante
5. Rolete (x2)
6. Cassete (x3)

Figura 17. Dispositivo de pesagem, sapata, legendado.

A elevação de cada roda da cama inicia-se pela aproximação da tesoura (2) à roda da cama e abertura dos dois braços da tesoura através de um botão (3) que translada a guilhotina libertando um veio escalonado que liga os dois braços da tesoura, seguindo-se o ajuste dos roletes (5) à roda. Caso não se consiga fazer um bom ajuste através do botão, pode-se recorrer a um volante (4) que permite um ajuste mais fino. Tendo-se os roletes bem posicionados,

tangentes à roda e paralelos entre si, exerce-se força na alavanca (1), elevando a roda do chão a uma determinada altura.

O gráfico 7 mostra para as rodas padronizadas, apresentadas no catálogo da TENTE, a distância destas ao solo após o processo de elevação acima descrito. (http://pt.tente.com)

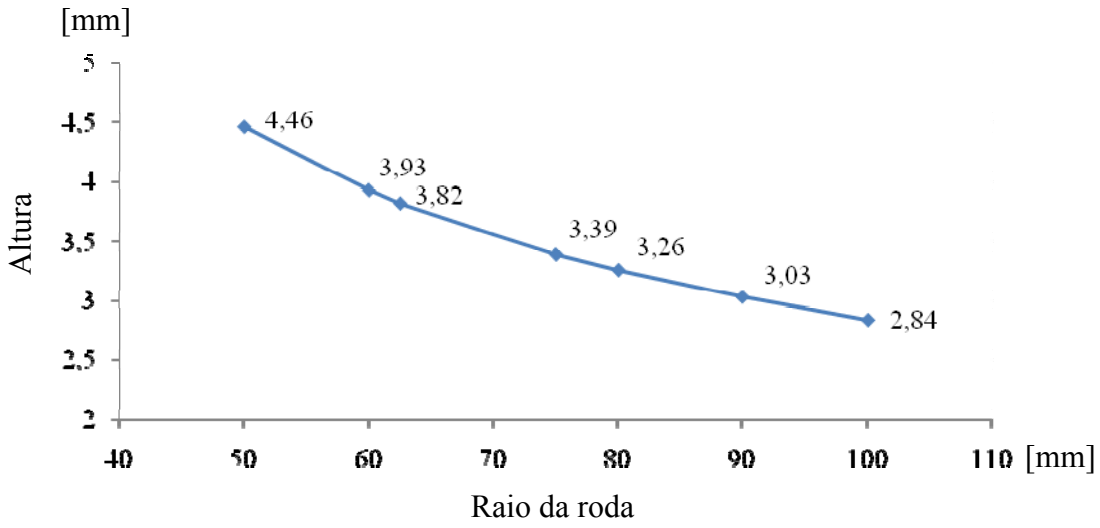


Gráfico 7. Distância das rodas ao solo em função do seu raio.

As quatro sapatas com electrónica dedicada transmitem os dados do peso, sentido pelos 12 elementos sensores, a uma consola de comando que faz o somatório e disponibiliza essa informação recolhida através de um LCD.

3.4 **Conclusão**

Após análise das características do sensor, verificou-se que este não satisfaz as necessidades pretendidas para este tipo de aplicação.

Com erros de linearidade e desvios que rondam os 5%, impedem que esta balança hospitalar tenha uma resolução aceitável. É necessário portanto fazer uma pesquisa de sensores que tenham menor distorção da informação que fornecem permitindo ter resoluções aceitáveis e que concorram com as restantes balanças já comercializadas. Contudo pode-se tentar um diferente tipo de condicionamento de sinal ou construir uma célula de carga dedicada, semelhante às que equipam as balanças electrónicas.

A substituição do tipo de sensor não implicará grandes mudanças na estrutura mecânica, pois ela já foi pensada para mudar o sistema sensorial, ou seja, para que o sistema de alojamento do elemento sensor seja modular.

No que toca ao princípio de funcionamento da estrutura mecânica, ela foi pensada para que pudesse acomodar rodas de diferentes dimensões e feitios e as elevasse do chão sem grande esforço do operador.

4 . Projecto Mecânico

4.1 Introdução

Um dos objectivos essenciais do Projecto final de alunos do Mestrado Integrado em Engenharia Mecânica é o saber projectar, e como tal, só na sequência do projecto é que realmente se tem contacto com as dificuldades e entraves constantes no desenvolvimento de um produto final viável. Para isso é essencial estabelecer alguns requisitos, conceber, trabalhar bem uma ideia e pré-dimensionar antes que se inicie o trabalho de modelação. É este trabalho de modelação, recorrendo ao programa *SolidWorks*, e de simulação dos esforços estáticos a que estão sujeitos os principais elementos mecânicos, utilizando o programa *COSMOSWorks*, que ocupará o desenrolar deste capítulo. A análise das tensões admissíveis, dos deslocamentos e das deformações fez apelo ao Método dos Elementos Finitos.

4.2 Dimensionamento mecânico

Todo o projecto de concepção mecânica tem, na sua essência, um largo período de amadurecimento que, partindo de princípios básicos, vai evoluindo por aproximações sucessivas para uma solução final de projecto. As figuras que se seguem ilustram parte do processo iterativo desenvolvido ao longo deste trabalho até à solução final. A figura 18 apresenta um sistema de carga do elemento sensor e a figura 19 um dispositivo de pesagem onde se destaca um pormenor da articulação central.

O dimensionamento mecânico transcrito neste ponto serviu de referência à modelação tridimensional de todo o dispositivo de pesagem, anteriormente referido como sapata. Como já havia sido dito, deste dispositivo fazem parte três sistemas: a tesoura, o sistema de elevação (constituído pela alavanca telescópica e pelo sistema de retenção do veio escalonado) e a cassete.

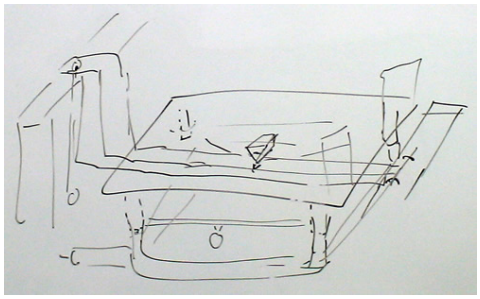


Figura 18. Estudo do sistema de carga do sensor.

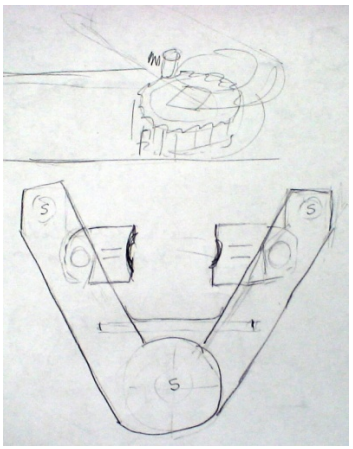


Figura 19. Estudo do dispositivo de pesagem.

Tendo em conta que a cama hospitalar possui uma estrutura rígida é de esperar que haja maior solicitação de três das quatro rodas devido a irregularidades do chão e da própria estrutura. Por conseguinte, nos cálculos desenvolvidos à posteriori e no estudo estático através do programa *COSMOSWorks*, considerou-se uma carga normal máxima aplicada em cada uma das sapatas de 170 kgf (1700 N), ou seja, um terço da carga máxima estipulada como objectivo deste projecto, 5000 N.

Na figura 20 pode ser visualizada a esquematização do processo de elevação da cama através da representação de uma roda (2), de dois roletes (1) e da aplicação da força normal *F*. A figura 21 apresenta dois polígonos de força. O (a) representa a carga aplicada na roda, distribuída pelos dois roletes e o (b) a decomposição da carga exercida num dos roletes.

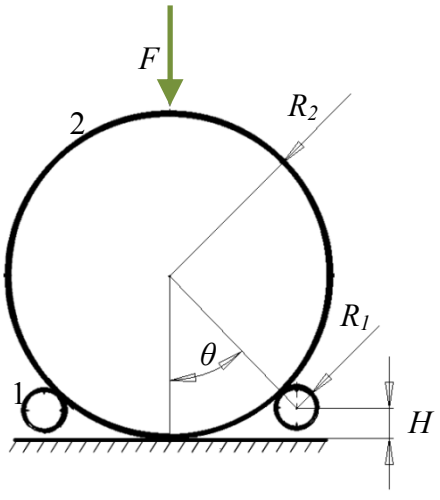


Figura 20. Esquematização do processo de elevação.

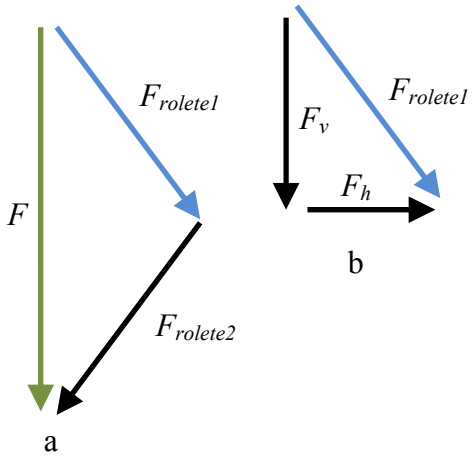


Figura 21. Polígonos de forças.

O raio dos roletes de encosto à roda (R_1) é de 10 mm, sendo R_2 o raio da roda, que pode variar entre 50 mm e os 100 mm. A altura H é a distância que vai desde o eixo do rolete ao solo, sendo o seu valor de 14 mm. Por fim θ refere-se ao ângulo entre o segmento de recta vertical, que passa no centro da roda, com o segmento de recta que une o centro da roda com o centro do rolete, ou seja o ângulo de encosto do rolete com a vertical. As dimensões atrás referidas, raio e altura dos roletes, surgem após a modelação 3D efectuada.

Os valores das forças verticais (F_v) e horizontais que aparecem ao nível dos roletes F_h e ao nível do veio escalonado F'_h, respectivamente, resultam da carga aplicada numa gama de rodas atrás consideradas sobre o dispositivo de pesagem, e estão apresentadas na tabela 7.

Tabela 7. Evolução das forças presentes no dispositivo de pesagem para diferentes tamanhos de rodas

R_2 [mm]	θ[°]	F_v [kgf]	F_h [kgf]	F'_h [kgf]
50	53,13	85	113,33	264,44
55	50,89	85	104,57	243,99
60	48,92	85	97,50	227,50
65	47,16	85	91,65	213,85
70	45,57	85	86,72	202,34
75	44,14	85	82,48	192,46
80	42,83	85	78,80	183,87
85	41,64	85	75,56	176,32
90	40,54	85	72,69	169,61
95	39,52	85	70,11	163,60
100	38,57	85	67,79	158,17

Tratando-se de uma elevação quase estática, o único parâmetro a ter em conta no sistema de elevação é o resultante do efeito peso da cama. Assim F_v é dado pela expressão

$$F_v = \frac{F}{2}[kgf] \qquad \text{(eq. 2)}$$

e a força horizontal descarregada sobre o rolete é dado pela expressão

$$F_h = F_v \times \tan\theta\,[kgf] \qquad \text{(eq. 3)}$$

com o θ dado pela expressão

$$\theta = \arccos\theta \frac{R_2 - H}{R_2 + R_1}[°] \qquad \text{(eq. 4)}$$

Finalmente a força ao nível do sistema de elevação, mais propriamente do veio escalonado, F'_h é dado pela expressão

$$F'_h = \frac{F_h \times a_1}{a_2}[kgf] \qquad \text{(eq. 5)}$$

em que a_1 e a_2 são as distâncias do ponto de articulação da tesoura às linhas de acção das forças como esquematizado na figura 22.

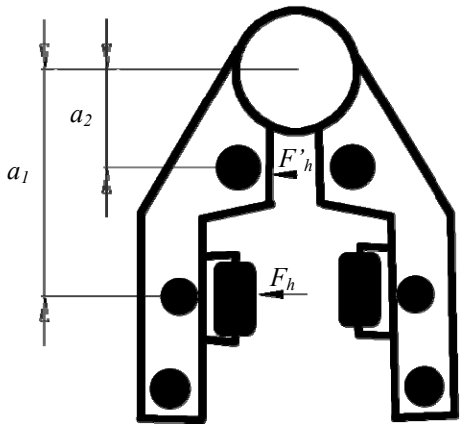


Figura 22. Aplicação das forças horizontais no sistema.

Estas forças resultam de se ter considerado R_1 = 10 mm, H = 14 mm e F = 170 kgf, como explicitado anteriormente, a_1 = 182 mm e a_2 = 78 mm, comprimentos resultantes da modelação tridimensional da tesoura.

No gráfico 8 pode ser visualizada, para maior clareza, a evolução das reacções horizontais exercidas no sistema de elevação em função dos diferentes raios das rodas.

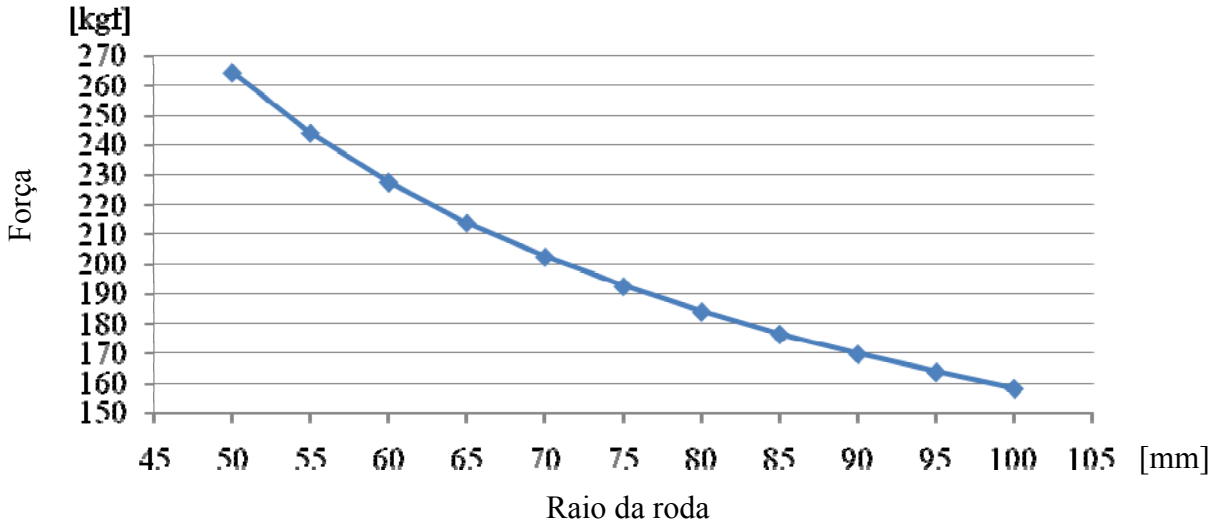


Gráfico 8. Reacção horizontal em função dos raios das rodas.

O comprimento da alavanca de actuação deve ser tal, que permita que o sistema seja operado sem grande esforço por parte dos utilizadores. Admite-se que 20 kgf seja a força máxima requerida ao utilizador. Como se pode constatar nos dados fornecidos anteriormente, a situação mais desfavorável ocorre quando intervém a roda de menor diâmetro. De toda a evidência, a situação que requer maior força ao utilizador para elevar a cama, corresponde à situação esquematizada na figura 23 que elucida quanto ao funcionamento da alavanca. Do equilíbrio de momentos no ponto *O* resulta, para o comprimento do braço da alavanca, $H_{alavanca}$, a equação 6

$$H_{alavanca} = \frac{F'_{h,máximo} \times 18}{F_{pretendida}} = \frac{300 \times 18}{20} = 270\,mm \qquad \text{(eq. 6)}$$

A dimensão de 18 mm surge da modelação 3D feita para este sistema, $F_{pretendida}$ representa a força máxima considerada confortável a exercer pelo utilizador e o $F'_{h,máximo}$ foi arredondado para os 300 kgf visando introduzir alguma força de atrito existente entre a cavilha ϕ10, a cavilha ϕ6, a manivela e bielas do conjunto alavanca telescópica e o deslizamento dos braços durante o fecho da tesoura. (confrontar Figura 16)

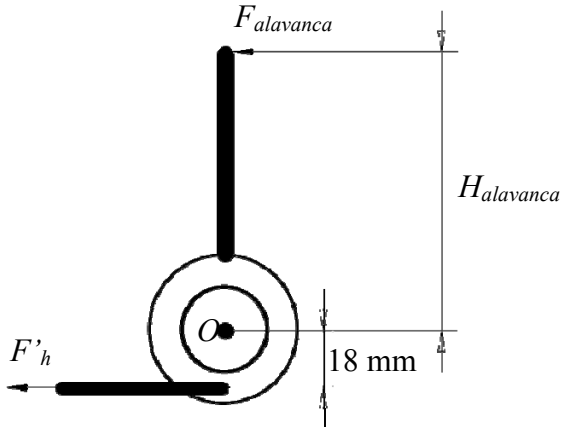
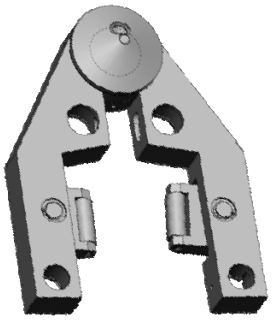


Figura 23. Esquematização do sistema de alavanca.

4.3 **Tesoura**

A tesoura é considerada como sendo o conjunto base que aloja os restantes sistemas mecânicos descritos posteriormente, tendo um peso de 5,2 kgf. Na figura 24 é apresentado o conjunto tesoura numa vista isométrica e numa vista explodida da estrutura mecânica.



Legenda:

1. Tampa roscada
2. Anilha de aço
3. Braço superior
4. Anilha INAFAG PAW62
5. Braço inferior
6. Veio ϕ18 mm (x2)
7. Suporte do rolete (x2)
8. Apoio roscado
9. Casquilho INAFAG PAP18 (x2)
10. Anilha INAFAG PAW18 (x2)
11. Veio do rolete (x2)
12. Casquilho INAFAG PAP10 (x6)
13. Tubo do rolete
14. Tampa do braço (x2)

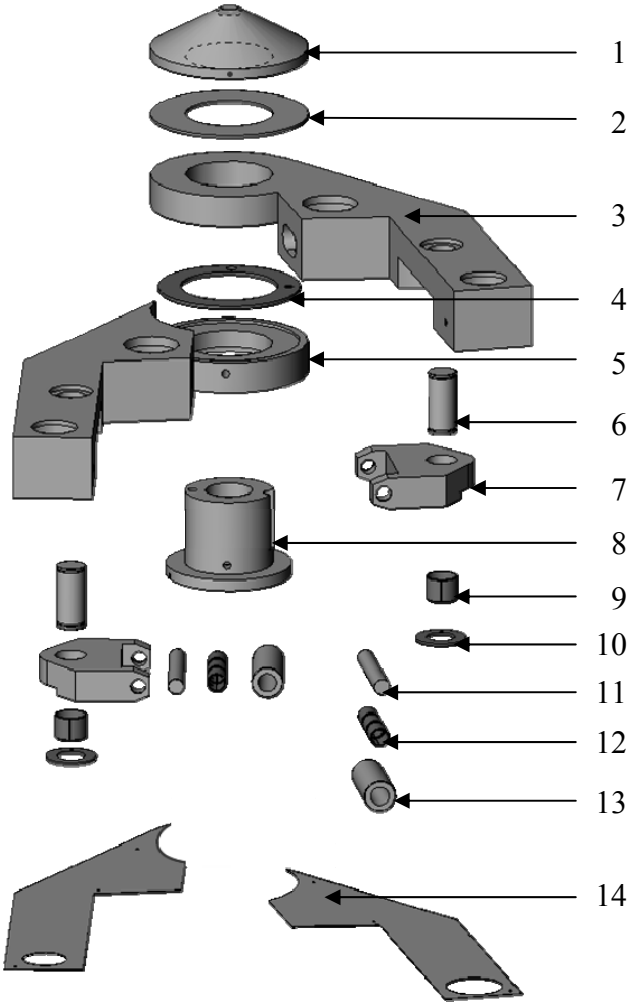


Figura 24. Vista isométrica e explodida da estrutura mecânica, tesoura.

A tampa roscada (1) é o componente onde será alojada a placa com a electrónica necessária ao condicionamento de sinal dos sensores que, juntamente com o apoio roscado (8) irá segurar o braço superior (3) e o inferior (5). Para garantir o posicionamento da tampa roscada e do apoio roscado, introduziu-se uma anilha rectificada (2) e para diminuir o atrito entre a superfície circular dos dois braços colocou-se uma anilha (4).

O suporte do rolete (7) é fixado aos braços através do veio (6) travado por um anel elástico em cada uma das suas extremidades. O veio do rolete (11), que está seguro ao seu suporte, alberga três casquilhos (12) revestidos a teflon, montados no tubo (13) que entrará em contacto com as rodas das camas hospitalares.

Por baixo da estrutura são colocadas duas tampas (14), que fecham os canais de condução de cabos.

Para o estudo estático deste conjunto, representado na figura 25, este foi sujeito a uma carga de 1700 N aplicada nas extremidades dos dois roletes. As restrições tiveram de ser colocadas estrategicamente de modo a conseguir manter este mecanismo estático. Foram colocados três pontos de restrição, correspondendo cada um deles a um apoio da tesoura. O apoio do lado esquerdo foi encastrado, restringindo a deslocação em qualquer uma das três coordenadas. No apoio central apenas se restringiu verticalmente, eixo Z, permitindo a deslocação no plano da articulação, com a intenção de se visualizar uma possível deformação na zona do centro. O apoio da direita foi restringido no eixo X e Z permitindo a abertura dos dois braços da tesoura.

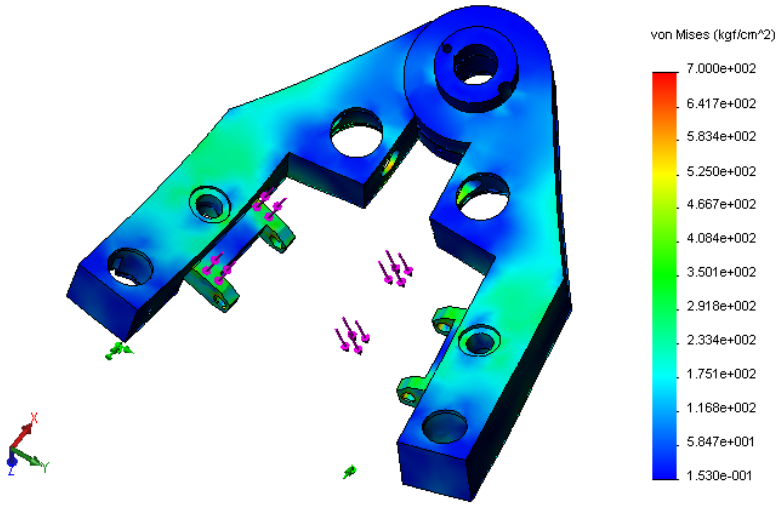


Figura 25. Distribuição das tensões na estrutura mecânica com um factor de escala de 100.

Como se pode verificar, as tensões nos braços da tesoura e no suporte dos roletes aproximam-se dos 6,5 kgf/mm^2, permitindo escolher um material em liga de alumínio da série 6000 com uma tensão de rotura de 31 kgf/mm^2, de forma a diminuir o peso total da estrutura. O suporte do rolete, é o componente cuja simulação sugere uma maior debilidade, contudo a tensão máxima que lhe está associada é de aproximadamente 8,5 kgf/mm^2, valor este que na realidade será menor pois o estudo não contemplou o rolete, elemento que é objecto de estudo mais à frente.

Os valores dos deslocamentos sofridos deste conjunto nesta simulação não transmitem a realidade devido ao encastramento feito no apoio da esquerda, contudo consegue perceber-se que não ultrapassará os 0,3 mm na zona de aplicação das forças, sendo a direcção Y a mais crítica. A figura 26 permite a visualização das deslocações sofridas pela tesoura.

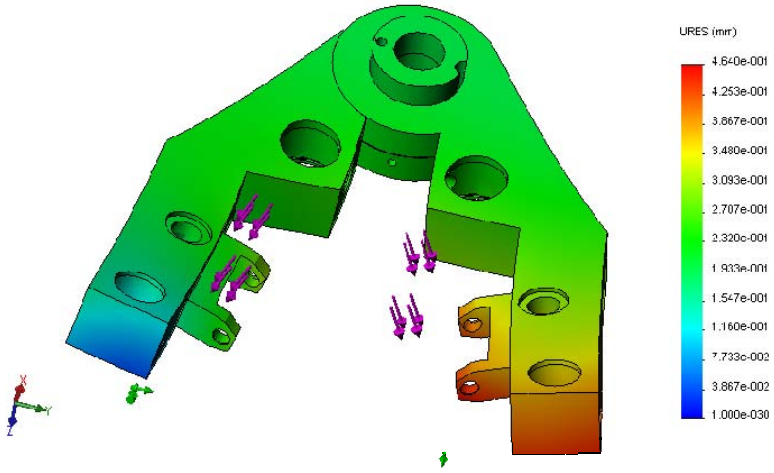


Figura 26. Distribuição das deslocações sofridas pela tesoura com um factor de escala de 100.

O rolete foi também estudado de forma pormenorizada. Na figura 27, são analisadas as tensões a que o rolete está sujeito, tendo sido aplicada uma força dupla da que ele virá a sofrer, 1700 N. A força foi aplicada no centro do tubo do rolete e as reacções foram colocadas no furo do suporte do rolete. Todas as outras peças, à excepção do suporte do rolete, foram ocultadas de forma a melhor se analisar o rolete.

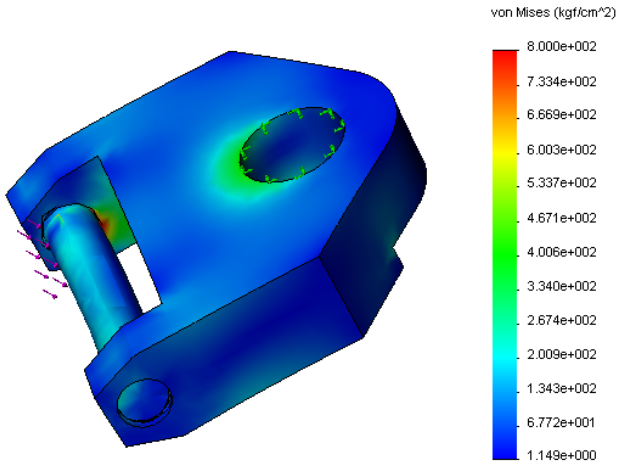


Figura 27. Distribuição das tensões no veio do rolete com um factor de escala de 112.

Os pontos onde há maior concentração de tensões são junto às faces interiores do suporte do rolete, atingindo uma tensão de aproximadamente 1,4 kgf/mm^2.

As deformações são desprezáveis rondando, na pior das hipóteses, as 7 centésimas do milímetro no centro do rolete.

Apesar do veio do rolete estar sobredimensionado ao nível das tensões e deformações, isso não implica a necessidade da sua optimização dimensional, porque por um lado temos que garantir uma superfície de encosto à roda que garanta a sua integridade, por outro lado é necessário manter a rigidez do veio de forma a não danificar os casquilhos, caso este se deforme, e finalmente o furo que apoia o veio do rolete, existente no suporte do rolete, necessita de uma boa superfície de contacto para evitar a sua plastificação.

4.4 Sistema de elevação

4.4.1 Alavanca telescópica

Este ponto compreende todas as peças relacionadas com a alavanca telescópica, estando descriminadas na figura 28.

Legenda:

1. Esfera roscada
2. Pino ISO 2338 (x2)
3. Tubo extensível 1
4. Tubo extensível 2
5. Mola de tracção
6. Manivela
7. Veio de união roscado
8. Cavilha ϕ10 mm
9. Cavilha ϕ6 mm (x2)
10. Biela (x2)
11. Base

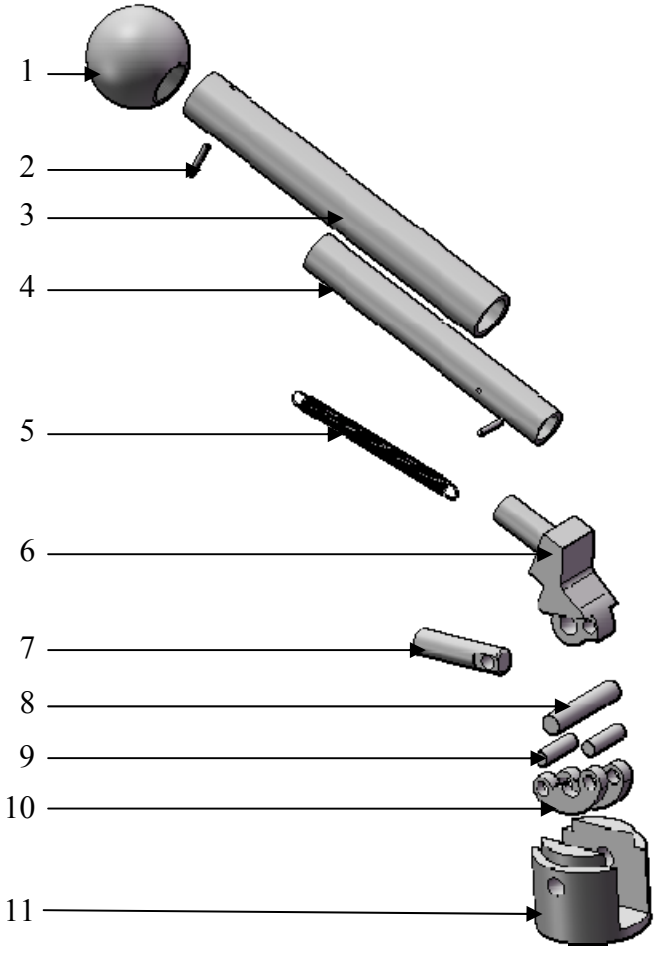


Figura 28. Vista isométrica e explodida da alavanca telescópica.

As peças referenciadas pelos números (1), (2), (3), (4) e (5), constituem o braço extensível. A mola (5), fixada através dos pinos (2), deve garantir a retracção do conjunto quando o braço do sistema está estendido. Para isso foi utilizada uma mola de diâmetro médio igual a 10 mm, 56 espiras e com um fio de diâmetro igual a 1,5 mm. O braço extensível é roscado à manivela (6), que por sua vez roda em torno da cavilha (8) montada nos furos da base (11). Ligada à manivela está ainda uma das cavilhas (9) que dá apoio às duas bielas (10), situando-se a outra cavilha (9) na extremidade oposta das bielas. Nessa mesma extremidade está acoplado um veio (7) que serve de ligação ao botão, sistema que será desenvolvido no ponto seguinte. O conjunto dos dois tubos que constituem a alavanca telescópica são impedidos de desmontar por limitação do curso imposta por um pequeno cabo que passa pelo seu interior e que não está representado.

Este conjunto é constituído por diversas peças de dimensionamento crítico onde se destacam as bielas e a manivela. A figura 29 apresenta o resultado do estudo realizado à biela.

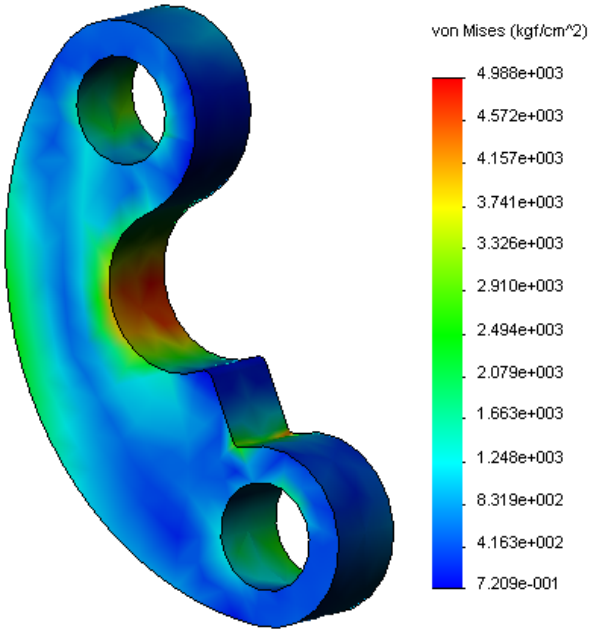


Figura 29. Distribuição das tensões na biela com um factor de escala de 1.

Para melhor se estudar as bielas, foram criadas duas peças paralelipipédicas com duas saliências cilíndricas, onde foram acopladas as peças a estudar. O estudo foi feito traccionando-se a biela à custa do encastramento de um paralelipípedo e da aplicação de uma força de 3000 N no outro, força esta justificada na equação 6 e texto adjacente. Na figura

anterior foram ocultas as peças de apoio, assim como uma das bielas com o intuito de melhor evidenciar a distribuição das tensões.

Verificam-se elevadas tensões na superfície côncava da peça, aproximadamente 50 kgf/mm^2, sugerindo assim, a utilização de materiais com maiores tensões limite elástico, tendo-se escolhido o aço DIN X 42 Cr 13 com tensão limite de rotura de 105/120 kgf/mm^2.

Quanto às deslocações sofridas, rondam o décimo de milímetro numa das suas extremidades, não sendo um valor significativo para o funcionamento em causa. A figura referente a este deslocamento pode ser encontrada no anexo B, juntamente com as restantes figuras que não aparecem neste capítulo.

A peça designada por manivela foi simulada na situação de elevação da cama. O resultado da simulação apresenta-se na figura 30. Para esta simulação carregou-se a extremidade da alavanca telescópica, na sua posição de extensão máxima, com 300 N, valor superior ao considerado nos cálculos, 200 N. Encastraram-se as terminações da cavilha onde roda a biela, ocultada na figura, e restringiu-se a cavilha de menor diâmetro onde são colocadas as bielas.

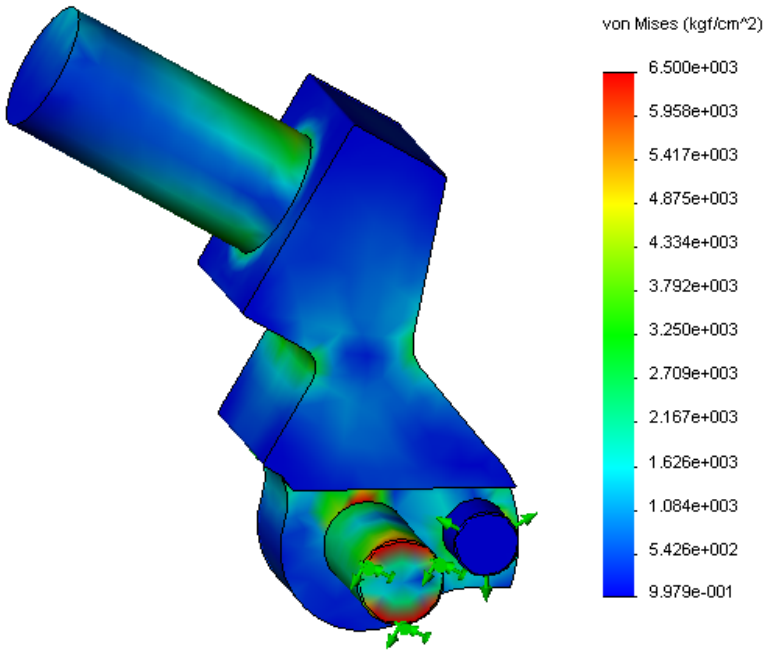


Figura 30. Distribuição das tensões na manivela durante a elevação da cama com um factor de escala de 1.

O caso mais desfavorável ocorre aquando da elevação da cama, correspondendo a uma força exercida na alavanca de cima para baixo. Durante o abaixamento da cama, o sentido da força será o mesmo, uma vez que o operador terá de controlar a descida de maneira a que esta seja suave, restando apenas o esforço necessário para vencer o ponto morto. Isto é, a recta (R) entre o centro das cavilhas ϕ6 (C_1 e C_2), situadas nos extremos de cada biela e consequentemente os esforços na ligação às cavilhas, passam acima do centro da cavilha ϕ10 (C_3) onde rola a manivela, como se pode ver na figura 31. Este ponto morto foi criado com a intenção de travar a abertura dos braços da tesoura quando esta está na posição de operacionalidade, ou seja, a manter as rodas da cama afastadas do solo.

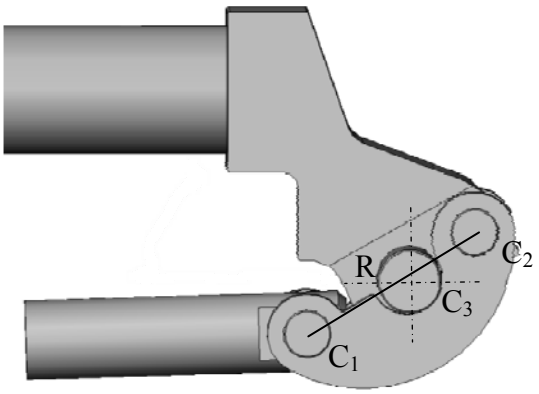
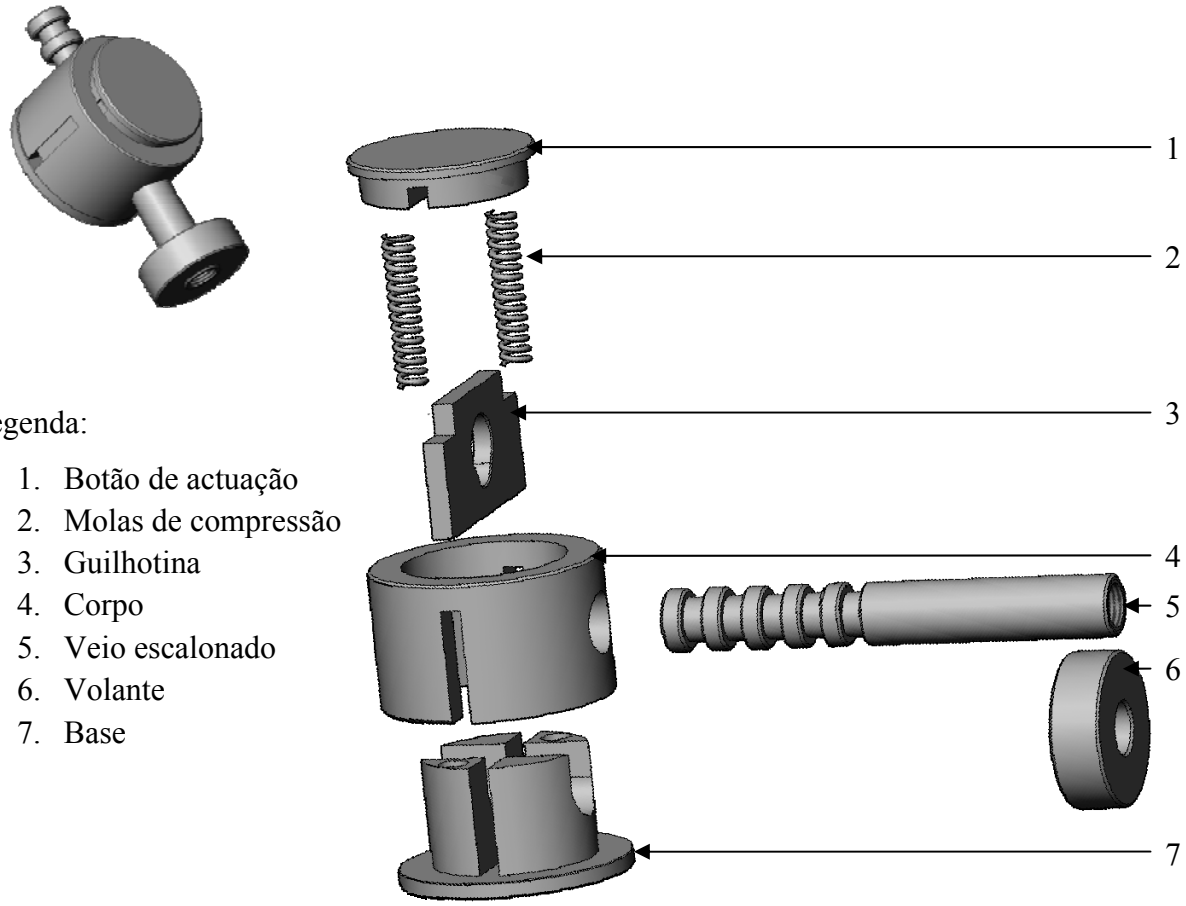


Figura 31. Ilustração do ponto morto.

Atendendo a que esta peça tem tensões instaladas que rondam os 70 kgf/mm^2, junto à cavilha de maior diâmetro, e que existe a possibilidade de má actuação por parte do utilizador não contemplada nos estudos, optou-se por utilizar um aço igual ao da peça estudada anteriormente, DIN X 42 Cr 13.

4.4.2 Sistema de retenção do veio escalonado

O conjunto de peças designadas por sistema de retenção do veio escalonado, estão representadas em vista isométrica e explodida na figura 32. Têm como função ajustar a abertura do sistema de forma a acomodar as rodas das diferentes camas hospitalares.



Legenda:

1. Botão de actuação
2. Molas de compressão
3. Guilhotina
4. Corpo
5. Veio escalonado
6. Volante
7. Base

Figura 32. Vista isométrica e explodida do sistema de retenção do veio escalonado.

O botão de actuação (1) permite soltar o veio escalonado (5), podendo assim abrir e fechar os dois braços que constituem a tesoura. Rigidamente ligada ao botão, a guilhotina (3) trava a translação do veio escalonado por intermédio da acção das molas de compressão (2), montadas com pré-carga. O corpo (4) constitui a peça que suporta todo o sistema. Por um lado limita o curso do conjunto botão/guilhotina por interferência com os rebaixos existentes nesta. Tem também como função o guiamento do veio escalonado. Permite finalmente fazer face às pequenas rotações que ocorrem durante a abertura e fecho da tesoura. Por baixo é colocada a base (7), travada à posteriori por um anel elástico de furo. Esta base tem a função de guiar as duas molas e conferir apoio à guilhotina que é uma das peças críticas neste conjunto.

As molas foram calculadas de forma a manter o topo do botão de actuação ao nível da superfície superior do braço da tesoura. Isto implica que a guilhotina tenha como posição natural o travamento do veio escalonado. Por outro lado foi necessário garantir a facilidade de operação através do dedo polegar, ou seja, o operador não deverá exercer demasiada força no

botão para que este solte o veio escalonado. A equação 7 [Giovanni77] foi a expressão utilizada no dimensionamento das molas:

$$F = \frac{f \times d^4 \times G}{8 \times n_u \times D_m^3} [kgf] \qquad \text{(eq. 7)}$$

Para o cálculo da força exercida por cada mola considerou-se uma deflexão f de 5 mm, um diâmetro de fio d de 1 mm, o módulo de resistência à torção G de 76,9 GPa, 14 espiras úteis (n_u) e um diâmetro médio D_m de 5 mm. A força resultante ronda os 3 kgf, sendo a força a exercer pelo utilizador de 6 kgf (duas molas).

Na figura 33 pode-se observar o resultado do estudo realizado à guilhotina e ao veio escalonado. Neste estudo foi aplicada uma força de tracção no valor de 2700 N, valor obtido por arredondamento das forças desenvolvidas para a roda de menor diâmetro, na ponta do veio escalonado e as forças de reacção foram colocadas em redor do corpo do sistema de retenção do veio escalonado.

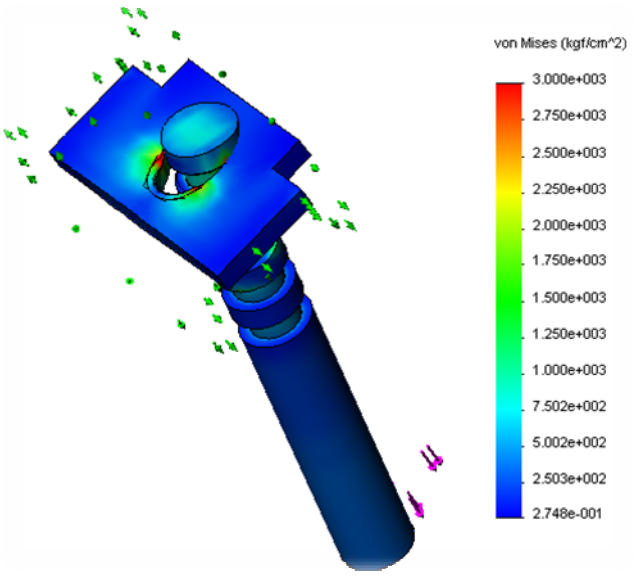


Figura 33. Distribuição das tensões na guilhotina e no veio escalonado com um factor de escala de 126.

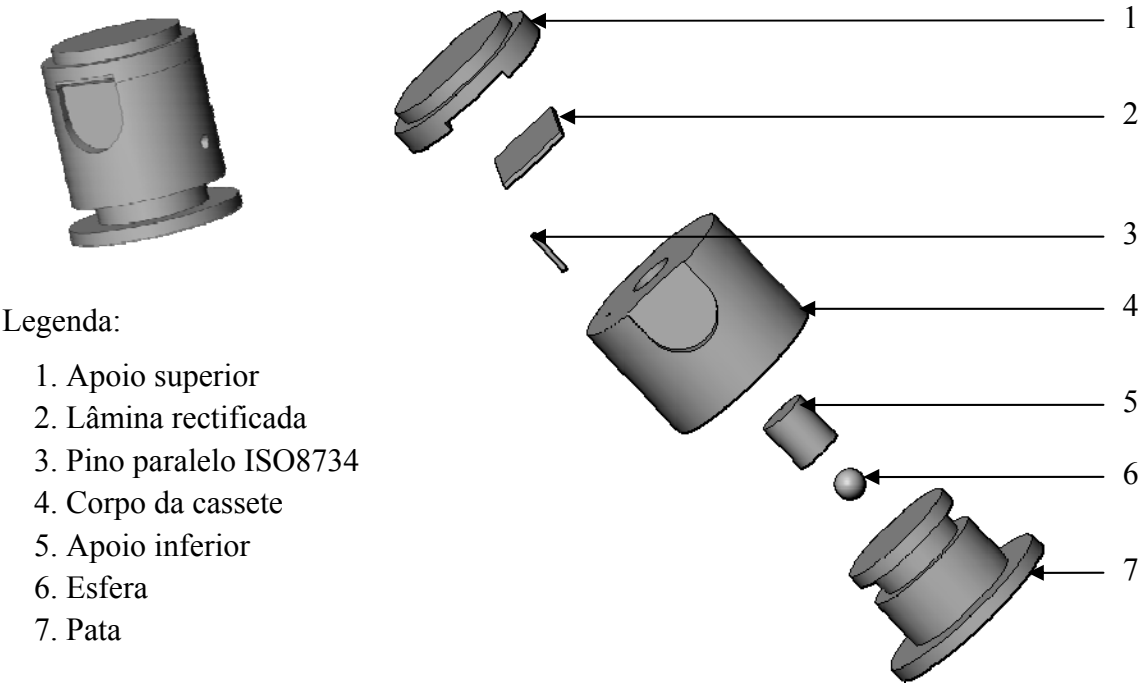
Como se pode verificar, existe uma concentração de tensões no contacto entre a guilhotina e o veio escalonado devido às deformações que ocorrem nas bordas do furo. Embora estas deformações possam provocar deformações plásticas localizadas, elas serão

desconsideradas pois não influenciam o normal funcionamento do dispositivo de pesagem. As peças de apoio à guilhotina e veio escalonado estão consideradas no estudo, porém foram ocultadas para permitir a melhor observação dos resultados. A tensão mais elevada localiza-se na face do veio escalonado que contacta com a guilhotina, apresentando um valor de aproximadamente 60 kgf/mm^2. Apesar das tensões atingirem tal valor naquela zona, o material escolhido para este conjunto foi um aço inoxidável DIN X 42 Cr 13.

As deslocações e deformações observadas neste conjunto são bastante pequenas, nunca passando a décima do milímetro no caso mais desfavorável, na extremidade do veio escalonado (ponto de aplicação da força).

4.5 **Sistema de alojamento do elemento sensor**

O sistema de alojamento do elemento sensor é apresentado na figura 34, através de uma vista isométrica e explodida.



Legenda:

1. Apoio superior
2. Lâmina rectificada
3. Pino paralelo ISO8734
4. Corpo da cassete
5. Apoio inferior
6. Esfera
7. Pata

Figura 34. Vista isométrica e explodida do sistema de alojamento do sensor, cassete.

O apoio superior (1) apoia a cassete ao braço do sistema, assim como acolhe uma lâmina rectificada (2), onde o elemento sensor, não representado na figura, é colado e

posicionado. O pino (3) é colocado para evitar a rotação entre o apoio superior (1) e o corpo da cassete (4), evitando desta forma, que as ligações do elemento sensor sejam danificadas. O carregamento do elemento sensor é conseguido por intermédio de um êmbolo denominado por apoio inferior (5), que possui a superfície de contacto rectificada, e atravessa o corpo da cassete (4). Na outra extremidade do apoio inferior aloja-se uma esfera (6) que garante um posicionamento correcto. Esta, por sua vez, é pressionada pela pata (7) que está em contacto com o solo.

O posicionamento do elemento sensor é a única dificuldade que a montagem deste sistema apresenta. Por indicação do fabricante, o sensor não deverá ser colado à sua base, uma vez que a existência de lacunas no filme de cola, impedirá o correcto carregamento do mesmo. Recorrendo a um perno roscado DIN 915, apresentado na figura 16, evita-se que o sistema se desmonte e rode em torno do seu eixo.

O apoio superior, elemento constituinte da cassete, deve possuir determinada rigidez de forma a apoiar correctamente o elemento sensor evitando, a introdução de erro na informação fornecida por este elemento assim como, evitar a sua degradação. A análise estática das tensões e dos deslocamentos, resultou da aplicação de uma força de 1700 N normal à superfície do apoio inferior, tendo-se restringido verticalmente a orla do apoio superior, como se pode ver na figura 35.

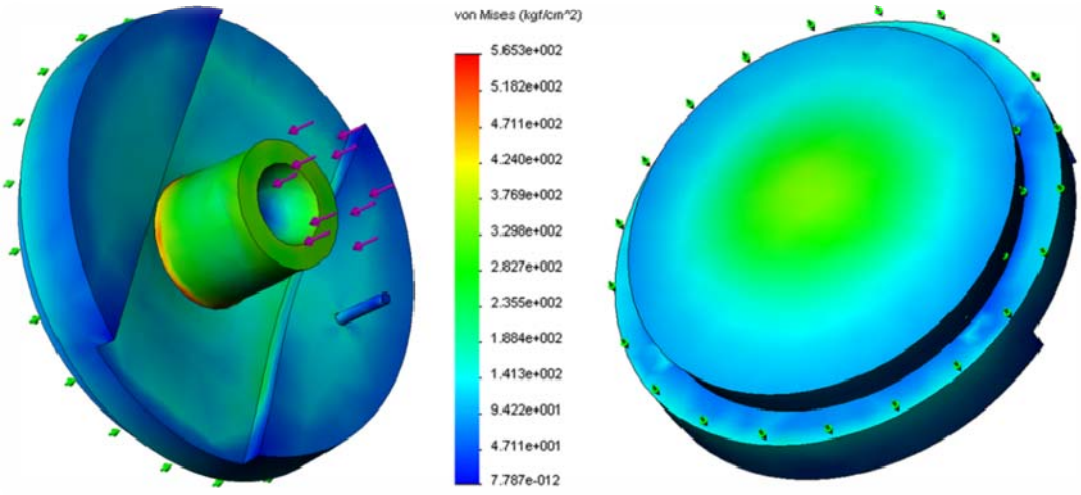


Figura 35. Distribuição das tensões na cassete com um factor de escala de 800.

Pode ser verificado na figura 36 que a zona onde o sensor está alojado sofre deslocamentos na ordem dos milésimos de milímetro, podendo ser considerada rígida.

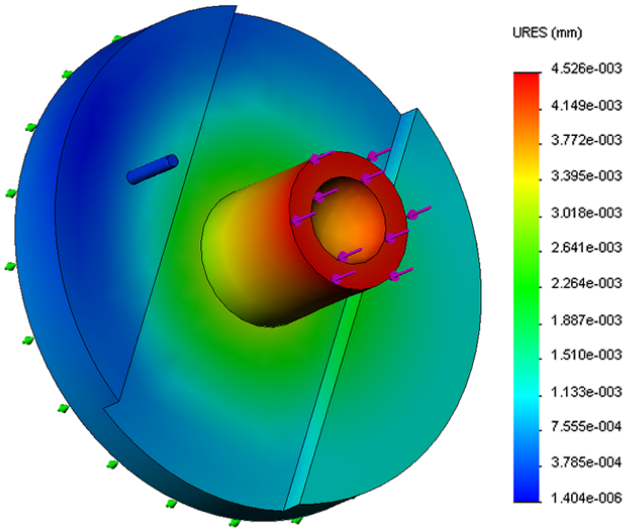


Figura 36. Distribuição das deslocações na cassete com um factor de escala de 1.

Dado que a peça denominada por pata apoia no chão à custa de uma superfície circular de diâmetro 40 mm e espessura 3 mm, fez-se um estudo às deslocações sofridas com o intuito de saber se ela não sofreria deformações prejudiciais. Para este estudo aplicou-se uma força de 1700 N na extremidade da base da pata e restringiu-se conforme se apresenta na figura 37. Constatou-se após o estudo que ela tem a tendência de revirar as suas bordas, contudo está suficientemente dimensionada para o cumprimento dos requisitos.

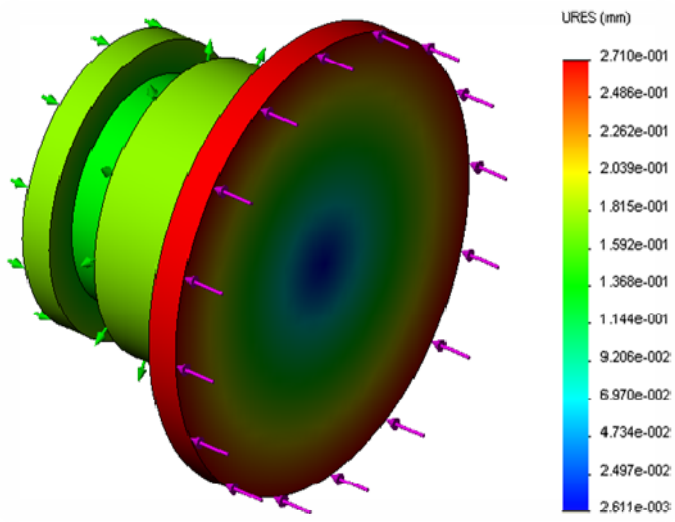


Figura 37. Distribuição das deslocações da pata da cassete com um factor de escala de 1.

Após análise destas figuras, representativas do estudo estático do conjunto cassete, optou-se por utilizar o aço inoxidável DIN 5 CrNiMo 17 12 2 com uma tensão de rotura de 62 kgf/mm^2.

4.6 **Conclusão**

A modelação apresentada permite-nos concluir que o princípio de funcionamento idealizado é exequível. Contudo, há necessidade de optimizar alguns componentes que estejam sobredimensionados.

As peças mais críticas são as bielas e a manivela, elementos constituintes da alavanca telescópica, tendo-se conseguido ultrapassar esta dificuldade recorrendo a um aço com uma tensão limite elástico superior aos aços mais vulgares, 105/120 kgf/mm^2.

A guilhotina e o veio escalonado são duas peças do sistema de retenção do veio escalonado que foram alvo de uma maior atenção uma vez que o veio provoca tensões superficiais significativas na guilhotina. Por conseguinte, podem surgir pequenas deformações plásticas na sua superfície que, depois das duas peças se acomodarem, não deverão trazer problemas ao seu funcionamento.

Para a escolha dos materiais nas diferentes peças teve-se em conta o seu tamanho e consequente peso, a solicitação durante a utilização, o acabamento superficial que determinado material permite e a degradação superficial a que está sujeito, como é o caso da oxidação.

O peso de cada sapata ronda os 6,5 kg. Embora o valor não pareça ser exagerado, ele não pode ser comparado com as outras balanças existentes pois, tal informação não é disponibilizada pela maioria dos fabricantes.

5 . Projecto Eléctrico/Electrónico e Informático

5.1 Introdução

Este capítulo inicia-se com uma pequena abordagem aos microcontroladores seguindo-se a escolha e caracterização dos diversos componentes electrónicos que constituem a placa de circuito impresso (PCB) da consola de comando e de cada uma das sapatas.

No ponto seguinte é dada uma breve explicação alusiva à placa de desenvolvimento, compilador, programa de escrita do código e das funcionalidades disponibilizadas ao operador.

Finalmente pode ser visto o entrosamento da electrónica e programação através da consola de comando construída.

5.2 Circuito electrónico

5.2.1 Caracterização dos principais componentes

O microcontrolador é um circuito integrado com funcionalidades que se assemelham a um computador contendo portos de entrada e saída, com o intuito de controlar dispositivos electrónicos. Têm como principais vantagens o seu baixo custo, rondando os 8 euros em média, baixo consumo energético, as dimensões reduzidas e eliminam a necessidade de utilização de determinados componentes externos. Como desvantagens, encontramos a baixa flexibilidade, pois não se pode alterar o seu interior e expansões limitadas às fornecidas de origem. São utilizados em diversos dispositivos electrónicos como o caso do teclado de um computador, relógios, máquinas de lavar e telefones, entre outros.

Para este trabalho utilizou-se um PIC (*Programmable Intelligent Computer* - Computador Inteligente Programável), produzido pela *Microchip Technology Inc.*. Esta opção deveu-se ao facto de haver a possibilidade de pedir amostras gratuitas e porque integra diversos periféricos necessários na realização do trabalho, como se poderá ver mais à frente.

Estes dispositivos existem numa ampla gama de modelos, que podem ser programados de forma a se ajustarem às necessidades de cada aplicação. Diferem no encapsulamento, na

memória, no número de entradas e saídas, na existência de conversores analógicos/digitais e num conjunto de saídas dedicadas, como por exemplo, saídas de PWM (*Pulse Width Modulation* - Modelação por Largura de Impulsos), utilizado para o controlo de motores eléctricos.

O PIC18LF4553 pertence à família da gama 18 dos microcontroladores de 8 bits. Assenta em memória Flash cuja informação faz parte integrante da sua designação através da letra F, designando a letra L a possibilidade de ser alimentado desde os 2,0 V até aos 5,5 V.

Este microcontrolador foi o seleccionado, para a aplicação em causa, pois possui 13 portas analógicas de 12 bits, entre outras características patentes na tabela 8.

Tabela 8. Características do microcontrolador

Memória do programa	Memória de Dados		I/O	12-Bit A/D a 50 ksps (canais)	MAX velocidade CPU (MHz)	Oscilador Interno (kHz)	nanoWatt
Flash (bytes)	SRAM (bytes)	EEPROM (bytes)					
32000	2048	256	35	13	48	8000 a 32	Sim

A figura 38 apresenta os pinos do microcontrolador utilizado caracterizados individualmente pelas diversas funções, podendo ser observados com mais pormenor na sua ficha técnica disponível no portal da *Microchip* (http://www.microchip.com).

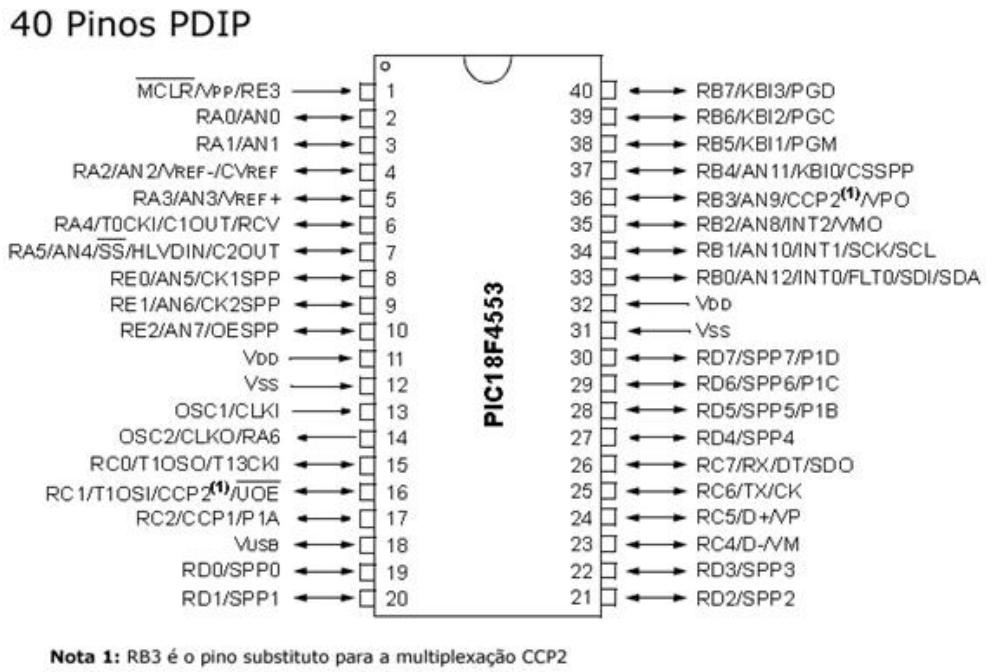


Figura 38. Pinos do microcontrolador.

Findada a caracterização do microcontrolador e justificação da sua escolha, apresenta-se na figura 39 o diagrama de blocos que ilustra a estrutura do projecto eléctrico/electrónico concebido. Faz-se notar que apenas contempla os componentes mais significativos. Este diagrama delimita dois grandes grupos: o circuito constante da consola (traço a cheio) e o circuito a implementar em cada uma das quatro tesouras (traço interrompido).

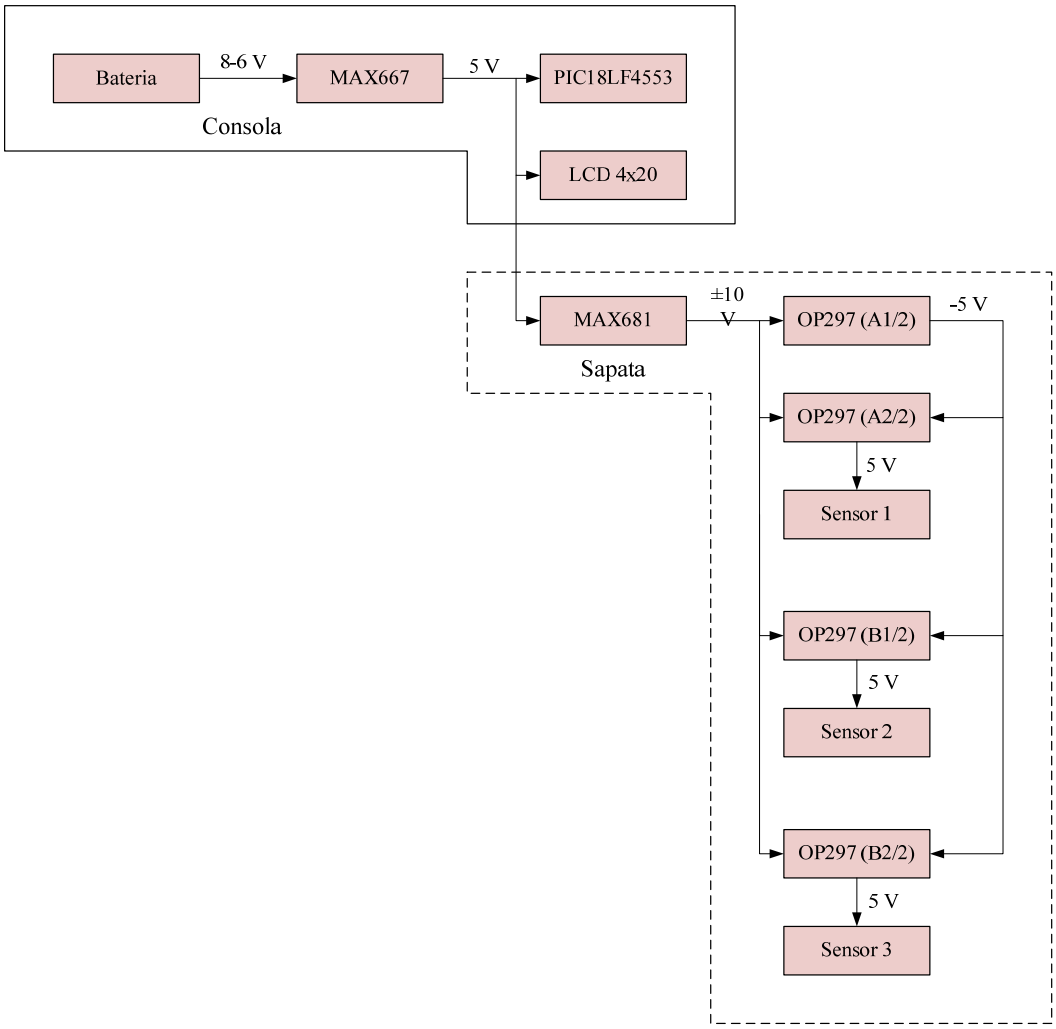


Figura 39. Diagrama de blocos dos circuitos eléctricos/electrónicos.

Todo o sistema será alimentado através de seis baterias AA recarregáveis, ligadas em série, que transmitem directamente toda a sua potência ao componente MAX667. Este componente é utilizado com o objectivo de regular a tensão a 5 V. Optou-se por utilizar seis baterias porque, por um lado, é necessário que o regulador tenha à sua entrada uma tensão superior à que vai regular, visto que a regulação é feita por abaixamento de tensão. Por outro

lado evita-se a utilização de uma bomba de carga, como o MAX660, que baixaria o rendimento do circuito, assim como tiraria vida útil às baterias.

Os 5 V do regulador são então distribuídos, de seguida, para o PIC18LF4553, para o LCD e para o MAX681.

O LCD é constituído por quatro linhas e vinte colunas e tem um consumo de 2 mA sem retroiluminação; caso se pretenda utilizá-la, o consumo sobe para os 400 mA.

O MAX681 é uma bomba de carga que, para uma entrada de 5 V, disponibiliza à sua saída ±10 V e uma corrente máxima de 10 mA que alimenta os amplificadores operacionais OP297 da *Analog Devices*. Cada integrado OP297 é constituído por dois amplificadores. Dos quatro amplificadores utilizados em cada sapata, um deles inverte a tensão de +5 V para -5 V e os outros três fazem a amplificação dos três sensores. Optou-se por este integrado por ter consumos mais baixos e maior estabilidade relativamente ao amplificador recomendado pelo fabricante do elemento sensor.

O consumo máximo da consola é de 6,6 mA sendo o de cada uma das sapatas de 9,7 mA, o que perfaz um consumo total máximo do sistema de 45,4 mA. Atendendo a que o sistema é alimentado por baterias, ligadas em série, de 2600 mAh estima-se que a autonomia da balança ronde as 55 horas.

5.2.2 Esquemas de ligações

Neste ponto são apresentados os esquemas eléctricos convertidos em placas de circuito impresso, incorporadas na consola e em cada uma das sapatas. Estes esquemas foram desenhados no programa *Eagle 5.0.0*, disponibilizado de forma gratuita na sua versão não profissional, cujas limitações prendem-se, essencialmente, com a dimensão da placa de circuito impresso que se pode desenvolver e com o seu número de camadas, 160 mm x 100 mm e duas camadas, respectivamente. Estas limitações não afectam as placas desenvolvidas pois não ultrapassam tais requisitos. Disponibiliza uma grande biblioteca de componentes e possibilita a criação de bibliotecas personalizadas caso não se encontre o componente pretendido ou um equivalente.

A figura 40 apresenta o esquema onde são mostradas as ligações entre alguns dos diferentes componentes, descritos anteriormente, assim como mais alguns, como é o caso de

resistências, condensadores, um LED e um *trimmer* que tem como função ajustar o contraste do LCD.

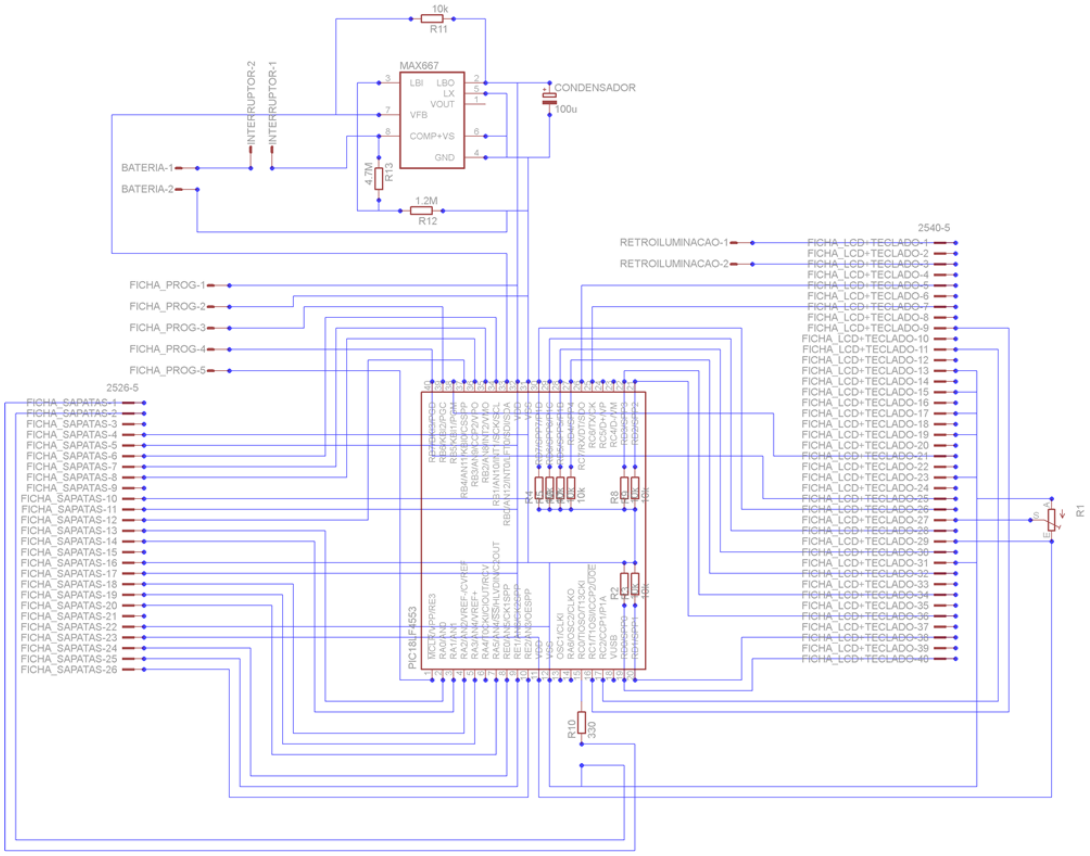


Figura 40. Circuito eléctrico constituinte da consola de comando.

A figura 41 contempla os componentes relacionados com o condicionamento de sinal utilizado, servindo de base à placa de circuito impresso a alojar em cada uma das quatro sapatas da balança. No Anexo E poderá ser vista uma ampliação das figuras 40 e 41.

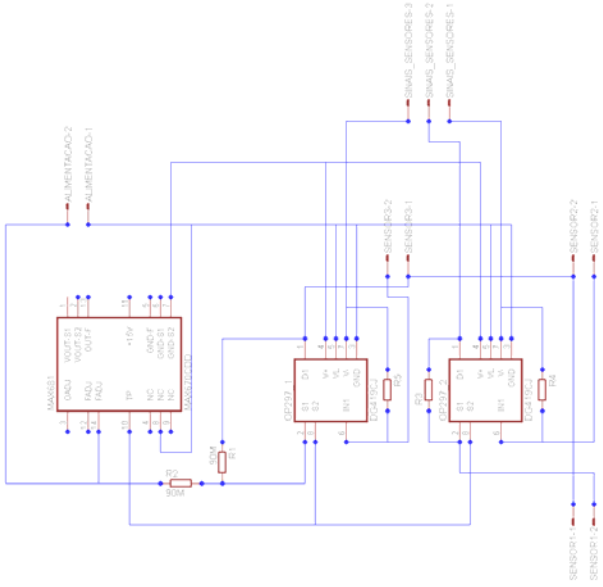


Figura 41. Circuito eléctrico constituinte de cada uma das sapatas.

Nas figuras 42 e 43 comparam-se as placas de circuito impresso (a) desenhadas no programa *Eagle* com as obtidas na realidade (b) e (c). No esquema, a face superior está representada a vermelho e a inferior a azul. A primeira placa possui as dimensões de 86 mm x 57 mm e a segunda de 36,5 mm x 33 mm.

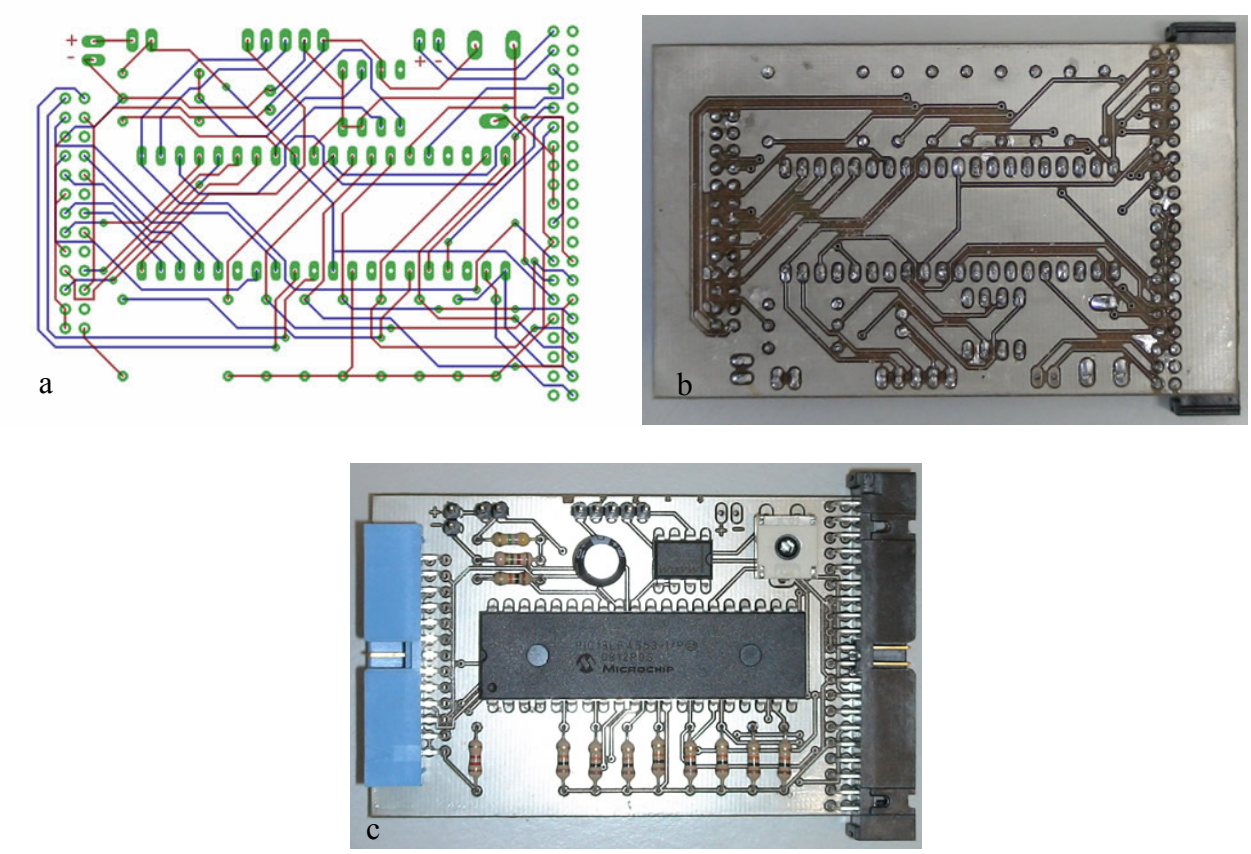


Figura 42. Placa de circuito impresso da consola.

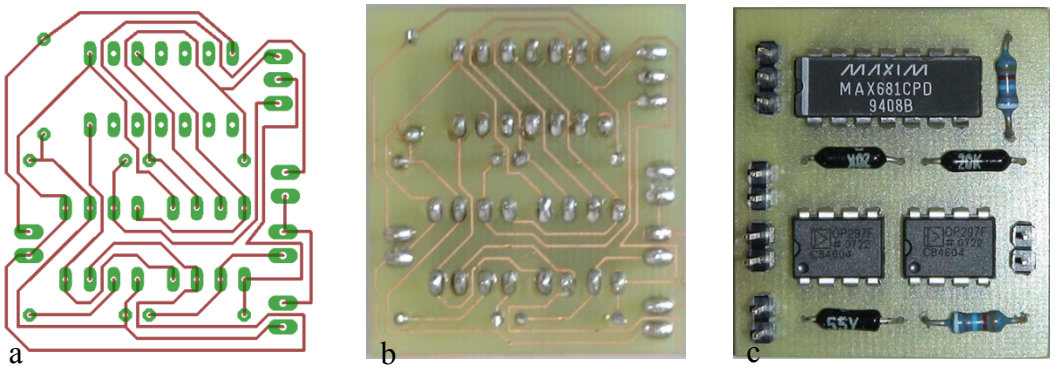


Figura 43. Placa de circuito impresso de cada uma das tesouras.

5.3 Programa desenvolvido

5.3.1 Sistema de desenvolvimento

A programação do PIC é feita recorrendo a um sistema de desenvolvimento da *MikroElektronica*, cujo pacote é constituído por três programas de escrita de código, cada um deles com uma determinada linguagem (C, Basic e Pascal), um programa de compilação e uma placa exterior de programação e teste.

A linguagem de programação utilizada foi a C e o programa denomina-se *MikroC*. É um programa que é constituído por diversas bibliotecas dedicadas aos PICs que, juntamente com o programa de compilação, *PICFLASH*, facilita a implementação de algumas funções. Por exemplo a instrução LCD_OUT(coluna, linha, texto) afixa no LCD o texto na posição indicada pelos dois primeiros argumentos da função; a Keypad_read() detecta a tecla pressionada ao nível do teclado. O aspecto do ambiente de trabalho é ilustrado na figura 44.

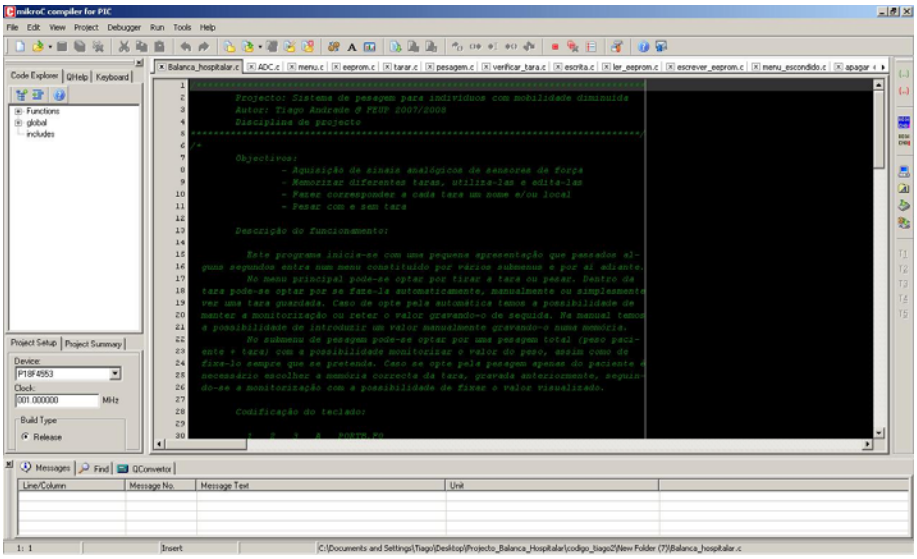


Figura 44. Aspecto do ambiente de trabalho do *MikroC*.

O compilador, cujo ambiente de trabalho se apresenta na figura 45, é um programa que interpreta e traduz o projecto do programa escrito em C num ficheiro hexadecimal, único formato reconhecido pelo microcontrolador. A sua instalação é feita na memória flash do microcontrolador fazendo apelo a 5 pinos, dois de alimentação (+5 V e 0 V), as portas RB6 e RB7 e a porta MCLR (*Master Clear* - Limpeza Mestre). Antes de passar o ficheiro hexadecimal, ele apaga a memória do microcontrolador colocando 12 V na entrada do MCLR.

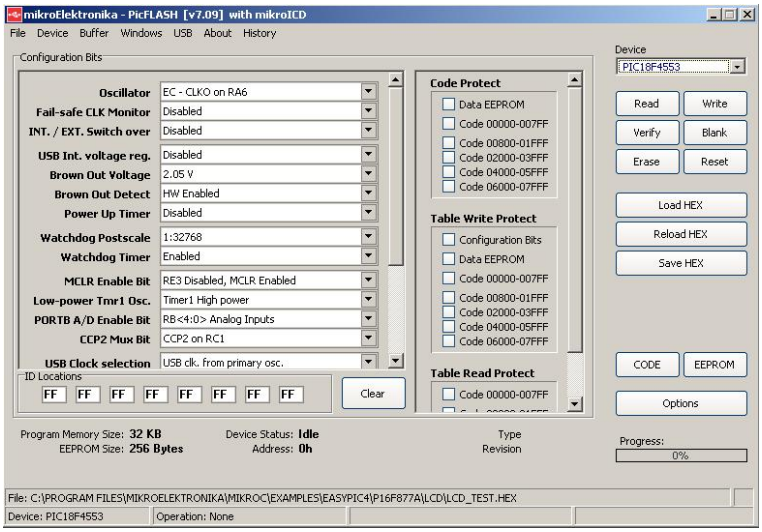


Figura 45. Compilador *PicFLASH*.

A placa de programação e teste, figura 46, é constituída por diversos terminais, admitindo assim PICs de 8, 14, 18, 20, 28 e 40 pinos. Possui comunicação RS-232 e USB 2.0, quatro monitores de 7 segmentos, um LCD de 2 linhas por 16 colunas, um LCD gráfico de 128x64 pixéis, alimentação externa ou por USB, 36 botões, 36 LEDs e diversos selectores que permitem diferentes configurações, assim como outras funcionalidades. Todos os pinos dos PICs estão ligados a conectores, permitindo assim associar expansões.

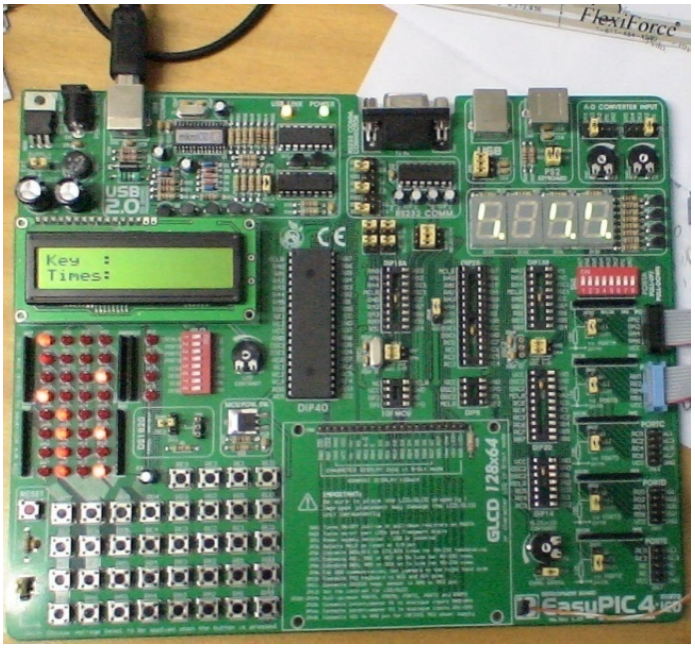


Figura 46. Placa de desenvolvimento

5.3.2 Código desenvolvido

Após terem sido estabelecidos os requisitos para o programa, atendendo à necessidade de adquirir e converter doze sinais analógicos, provenientes dos doze sensores piezoresistivos, e à simplicidade da comunicação com o utilizador, listaram-se as várias funções necessárias a implementar.

Na figura 47 é apresentado o teclado utilizado neste sistema, a fim de facilitar a compreensão do programa, nomeadamente nas transições entre as suas principais tarefas. O diagrama de blocos simplificado da estrutura principal do programa instalado no PIC é esquematizado na figura 48.

Figura 47. Teclado utilizado na consola.

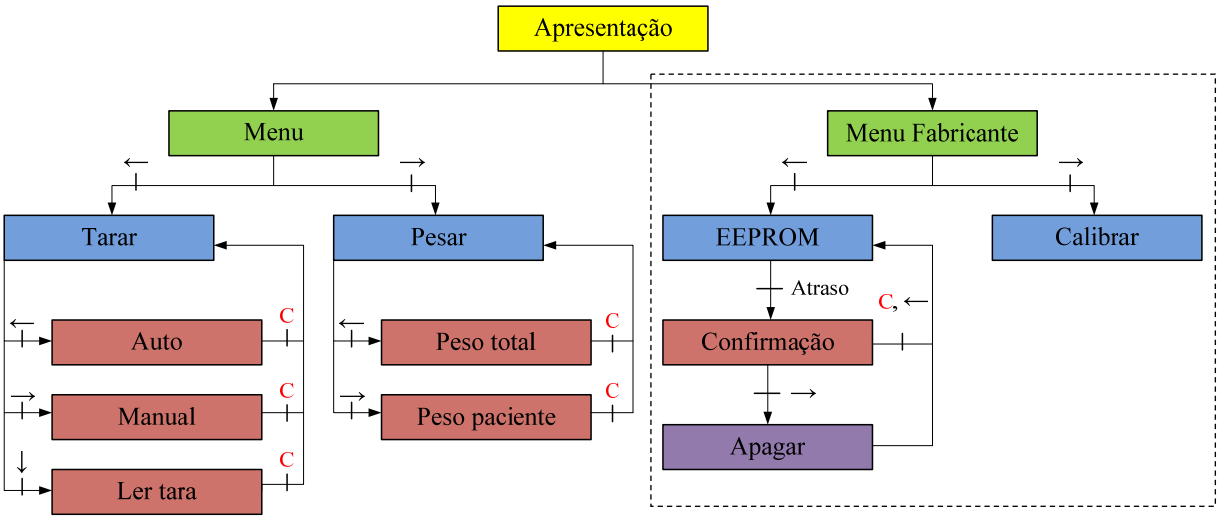


Figura 48. Diagrama de blocos simplificado do programa desenvolvidos.

O código escrito é constituído por 23 parcelas de código denominadas de rotinas, subrotinas e procedimentos.

O programa é iniciado com a rotina principal (*main*), onde corre o procedimento "config", que tem como principal função definir as portas do PIC como entradas e/ou saídas e inicia-las a zero, configurar os registos da componente analógica e de oscilação do cristal interno, que embora oscile a 8 MHz, está a ser utilizado a 1 MHz através da divisão de frequência previamente configurada, fazer a inicialização do teclado e do LCD, terminando com instruções que dão lugar à apresentação do dispositivo desenvolvido. Após este procedimento, e mediante uma determinada sequência de teclas, entra-se numa pequena subrotina, chamada de "Menu_Fabricante", reservada apenas para o actual programador, tendo duas funções: a limpeza da memória *eeprom*, rotina já implementada, e a calibração dos sensores, rotina a implementar. Caso contrário entrará na subrotina "Menu".

Dentro da subrotina "Menu", dependendo da escolha do utilizador, tem-se acesso a duas rotinas que são o pilar de todo este programa, as rotinas "Tarar" e "Pesar".

No Menu "Tarar" pretende-se memorizar valores de taras para serem utilizadas à posteriori no Menu "Pesar". Nesta rotina o utilizador pode optar por fazer uma tara automática, uma tara manual ou simplesmente visualizar as taras guardadas em memória, tendo a possibilidade, nesta fase, de editar os campos que contêm a informação do valor da tara e da identificação associada como traduz a figura 49.

Figura 49. Visualização de uma memória das memórias.

Na tara automática, o sistema apresenta o somatório dos valores lidos nos 12 elementos sensores. Dando a ordem de memorização do valor da tara, o sistema pesquisa as memórias livres na *eeprom*, disponibilizando a informação no LCD para que o utilizador faça a selecção desejada.

Na tara manual, o sistema permite ao utilizador digitar o valor da tara que pretende, seguindo-lhe o mesmo processo de memorização descrito anteriormente. A figura 50 é

constituída por um diagrama de blocos que pretende dar uma ideia das funções disponibilizadas no Menu "Tarar". A descrição de cada rotina poderá ser consultada no anexo A através dos comentários colocados ao longo do código. A designação associada a cada subrotina é suficientemente expedita para compreender o que cada uma executa.

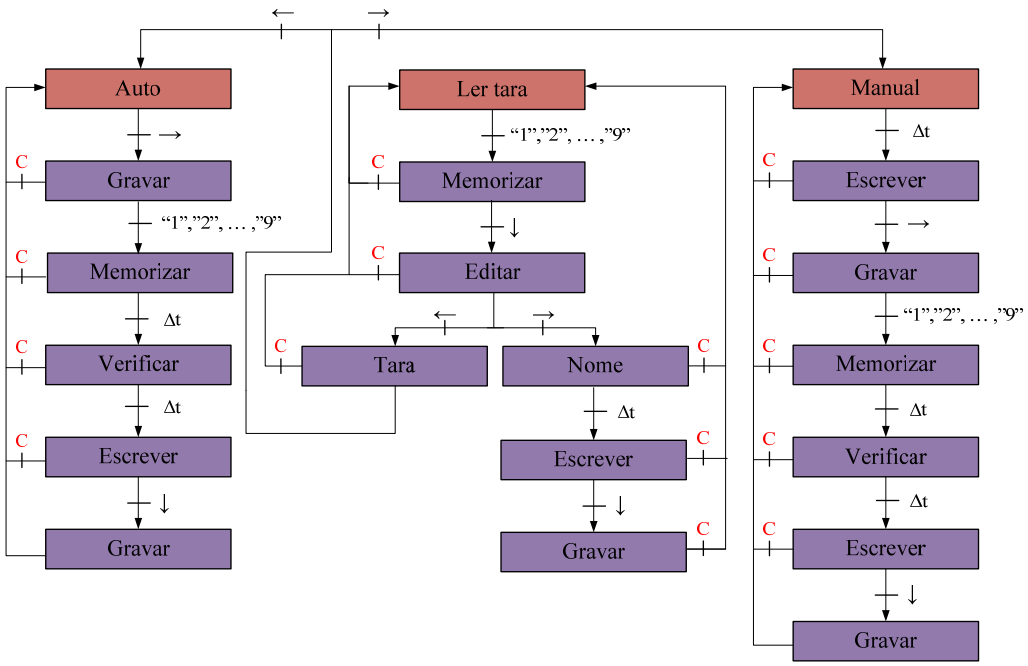


Figura 50. Diagrama de blocos simplificado do menu "Tarar".

O Menu "Pesar" subdivide-se em dois menus. O primeiro permite monitorizar o peso total, ou seja, o peso do paciente mais o peso da cama e equipamento associado e no segundo o peso do paciente, isto é, o peso total com a subtracção da tara memorizada anteriormente.

No menu referente ao peso do paciente, deve definir-se, previamente, um valor crítico para a variação do peso a monitorizar e o valor da tara anteriormente memorizada. As configurações fornecidas numa primeira pesagem, variação crítica e tara, são memorizadas para que na segunda pesagem do mesmo indivíduo, a que corresponde uma mesma cama, não seja necessário introduzir novamente os mesmos dados. Caso a balança seja utilizada numa outra cama, há a possibilidade de mudar essa configuração. A variação crítica introduzida tem como objectivo alertar o pessoal hospitalar para a alteração do peso, sendo isso feito através de um LED. Este poderá ser apagado através da tecla A.

Durante a monitorização, seja através do menu "Peso total" ou "Peso paciente", há possibilidade de reter o valor lido no LCD, retomando-se de seguida a monitorização por ordem do operador.

Na figura 51 é mostrado um diagrama de blocos que caracteriza as diversas fases do Menu “Pesar”.

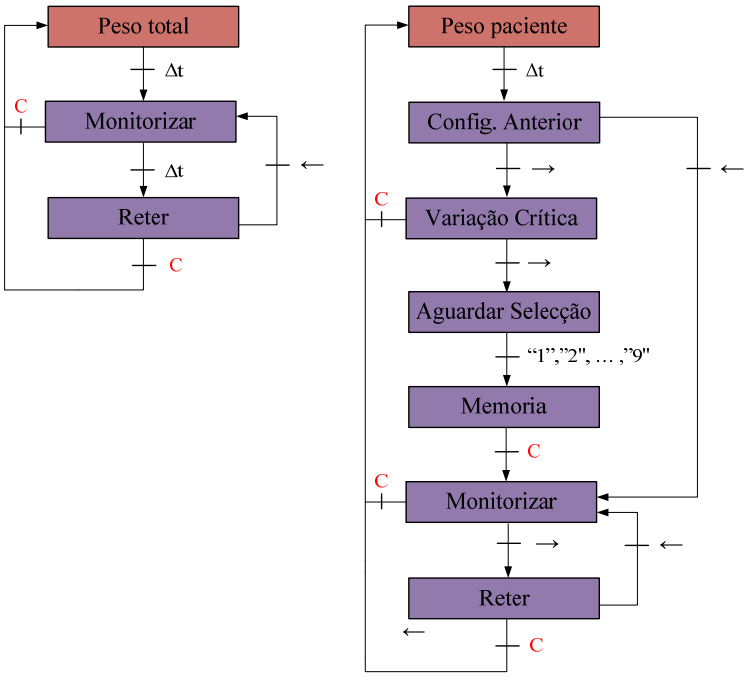


Figura 51. Diagrama de blocos simplificado do menu pesar.

O código associado ao funcionamento do teclado, foi desenvolvido de maneira a que este tivesse um comportamento semelhante ao de um telemóvel. A tabela 9 ilustra as funcionalidades de cada tecla, e o seu valor quando se utiliza na escrita.

Tabela 9. Funções e valores associados a cada tecla.

TECLA	FUNÇÕES	VALORES
1	escrita e selecção de memória	1 A B C
2	escrita e selecção de memória	2 D E F
3	escrita e selecção de memória	3 G H I
A	desligar alarme e escrita	- * _ +
4	escrita e selecção de memória	4 J K L
5	escrita e selecção de memória	5 M N O
6	escrita e selecção de memória	6 P Q R
.	escrita	. , : ;
7	escrita e selecção de memória	7 S T U
8	escrita e selecção de memória	8 V W X
9	escrita e selecção de memória	9 Y Z Ç
C	cancelar	
0	escrita	0 ? ( )
←	navegar menu e escrita	apagar
↓	navegar menu e escrita	
→	navegar menu e escrita	espaço

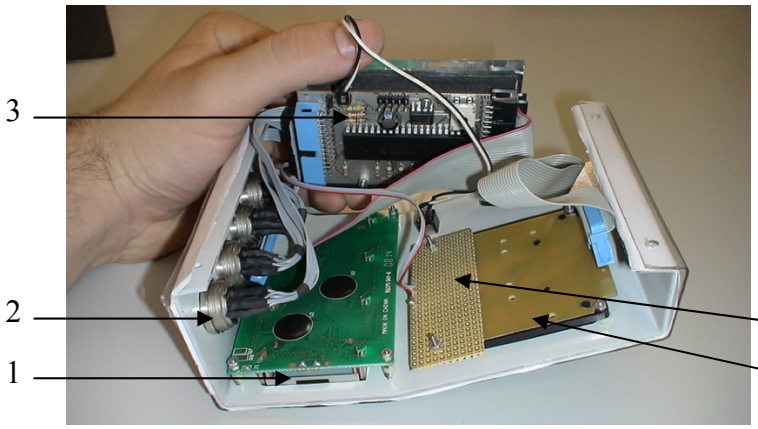
5.4 **Consola de comando**

Após a montagem dos componentes na placa de circuito impresso, da programação do microcontrolador e da adaptação de uma caixa metálica adquirida, procedeu-se à integração de tudo com o objectivo de obter uma consola de comando amigável. Com ela, o utilizador poderá executar as diversas operações referidas no ponto anterior.

A figura 52 apresenta uma fotografia (a) da vista de cima da consola. Nela pode ver-se o LCD e o teclado, elementos de interacção entre a balança e o operador, o interruptor Ligar/Desligar e o LED de alarme. Na fotografia (b), vista de trás, é possível observar quatro fichas onde são ligados quatro cabos. É através destes cabos que é feita a comunicação desde os sensores até à consola de comando. Por último, na fotografia (c), é ilustrada a vista de frente da consola, onde são fornecidas diversas informações identificativas.

Figura 52. Vistas de cima, trás e frente da consola de comando.

O interior da consola é mostrado pela figura 53 à qual se associou uma legenda.



Legenda:

1. LCD
2. Ficha (x4)
3. Placa de circuito impresso
4. Suporte do LED
5. Teclado

Figura 53. Interior da consola de comando.

Na figura 54 é possível ver em pormenor a ficha de programação do microcontrolador montada na consola de comando ao nível do suporte das pilhas (a), assim como o cabo (b) utilizado para programar. A programação do microcontrolador, integrado na consola, é na mesma feita recorrendo-se à placa de desenvolvimento atrás referida. (confrontar Figura 46)

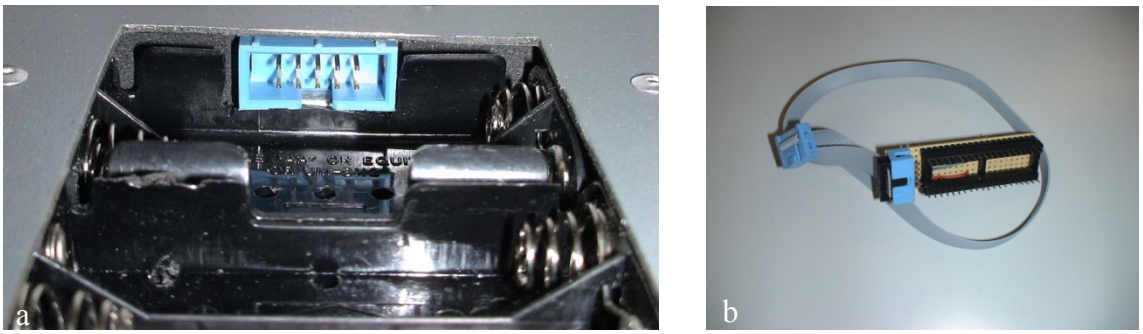


Figura 54. Elementos para programação do microcontrolador montado na consola de comando.

5.5 **Conclusão**

A incorporação deste microcontrolador beneficiou a configuração de todo o circuito pois facilitou a redefinição de tarefas sem alteração de dispositivos físicos. Isto é, através da programação conseguiu-se facilmente baixar o custo de implementação porque conseguiram-se funcionalidades que com outros dispositivos seria mais complicado, mais

caro e possivelmente mais limitado. É graças a ele que toda a electrónica analógica se simplificou, deixando de ser necessária a utilização de conversores A/D externos, componentes multiplexadores e cristal externo.

Surgiu a dificuldade de introduzir uma seta com sentido descendente no LCD, tendo sido contornado o problema utilizando a letra "V". Contudo existe a possibilidade de criar novos caracteres programando o microcontrolador do LCD, no entanto essa tarefa será reportada para os trabalhos futuros.

Os menus implementados para utilização da consola de comando da balança são bastante simples e intuitivos, permitindo uma rápida familiarização, por parte do utilizador, com o funcionamento do dispositivo. Contudo há sempre a possibilidade de consultar o manual do utilizador que conta no anexo D.

6 . Conclusão Geral e Sugestões de Trabalhos Futuros

6.1 **Conclusões**

O objectivo deste trabalho é o de desenvolver um sistema de pesagem para indivíduos com mobilidade diminuída que responda aos seguintes requisitos:

- Portátil, de funcionamento de tipo electrónico com alimentação autónoma;
- Orientado para medição em camas hospitalares com rodas, eventualmente extensível a outros tipos de camas ou cadeiras com rodas;
- Capacidade de carga total até 5000 N, incluindo tara;
- Sistema centralizado de avaliação com leitura digital, permitindo compensação manual ou automática da tara;
- Manuseabilidade simples do sistema e sem recurso a mais do que um operador;
- Baixo custo de produção e reduzida necessidade de manutenção.

Para satisfazer os objectivos propostos para este trabalho foram realizadas diversas pesquisas, utilizando motores de busca informáticos, sobre os dispositivos de pesagem disponíveis no mercado, com vista a encontrar as suas principais características, pondo em relevo as suas fragilidades.

Posteriormente idealizou-se um sistema mecânico que, depois de diversos processos iterativos, foi modelado e dimensionado. Após o dimensionamento concluiu-se que o dispositivo desenvolvido, embora possua sistemas delicados a nível estrutural, cumpre os requisitos propostos.

Com o peso rondando os 6 kgf por sapata, pode-se considerar que a balança desenvolvida não coloca problemas substanciais à sua manipulação, apresentando flexibilidade para a sua utilização em meios hospitalares.

No que se refere ao projecto eléctrico/electrónico e informático, os circuitos desenvolvidos foram implementados e testados com sucesso, tendo originado o desenvolvimento de uma consola de comando que centraliza toda a informação.

Os sensores utilizados revelaram-se inadequados para a aplicação em causa, uma vez que são incapazes de satisfazer os requisitos de exactidão e repetibilidade exigido pelas normas que regem este tipo de aparelhos.

6.2 **Sugestões para trabalhos futuros**

Visando finalizar o trabalho apresentado neste relatório, propõem-se as seguintes sugestões de evolução:

- estudar outro tipo de sensorização,
- desenvolver um sistema que permita articular os roletes por forma a manterem-se paralelos entre si,
- alterar ligeiramente a forma dos braços de maneira a aumentar a gama de diâmetros de rodas admissível e diminuir o peso da estrutura,
- elaborar os desenhos de definição, apartir da modelação 3D, que levarão à construção do protótipo,
- projectar uma consola mais leve e ergonómica,
- optimizar a programação e acrescentar a rotina de calibração dos elementos de medição,
- adicionar a função de carregamento de baterias e alimentação externa,
- aprofundar o estudo das normas e validar todo o sistema,
- calibração final da balança,
- elaborar uma versão com motorização do fecho dos braços da tesoura e com comunicação "sem fios" que permita controlar todo o sistema.

7 . Referências Bibliográficas

7.1 **Livros**

[Giovanni77] Giovanni Manfé, Rino Pozza, Giovanni Scarato. *Desenho técnico mecânico*. Hemus livraria editora limitada, p – 207, 1977.

José Manuel de Simões Morais. *Desenho Técnico Básico, Desenho de Construções mecânicas 3*. 22ª Edição, Porto Editora, Abril de 2002.

Fábio Pereira. *Microcontroladores PIC programação em C*. 1ª Edição, Editora Érica Ltda., 2003.

7.2 **Páginas da *internet***

Os portais que a seguir se apresentam foram revistos no dia 15 e 16 de Julho de 2008.

- Saúde

  http://www.infarmed.pt

  http://www.copacabanarunners.net/

  http://www.roche.pt/portugal/

  http://www.adexo.pt/

  http://www.who.int/

  http://www.scielo.br/

- Equipamentos

  http://www.instructables.com/id/S3FIZBZFDO2068H/

  http://ts.nist.gov/WeightsAndMeasures/h44-03.cfm

  http://www.coventryscale.co.uk/site/

  http://www.srinstruments.com/Scales/

  http://intercromo.com/montra/product_info.php/products_id/252

  http://www.scale-tronix.com/

https://www.seca-online.com/

http://www.healthchecksystems.com/

- Patentes

http://www.uspto.gov/

https://www.patentfamily.de/

http://www.freepatentsonline.com/

- Documentação

http://ts.nist.gov/WeightsAndMeasures/h44-03.cfm

http://www.hc.unicamp.br/servicos/emtn/

http://www.ipac.pt/docs/documentos.asp

http://www.oiml.org/

http://www.ipq.pt

http://dre.pt/

http://www.servico-metrologia.com

http://ts.nist.gov/WeightsAndMeasures/h44-03.cfm

http://www.macdata-ba.com.br

http://www.amla.pt/metrologia.html

- Rodas para camas hospitalares

http://pt.tente.com/

http://www.rodalges.pt/docs/rodas.html

- Camas

http://www.primuscareajudas.com/

http://www.fotosearch.com.br/fotos-imagens/cama-hospital_2.html

http://www.equipamentoshospitalares.com.br/cama_hospitalar.asp

- Sensor

http://www.tekscan.com/flexiforce.html

- Mecânica / Electrónica / Programação

http://medias.ina.de/medias

http://www.igus.pt/

http://pt.rs-online.com/web/

http://pt.farnell.com/

http://www.aquarionet.com/

http://www.maxim-ic.com/

http://www.ti.com/

http://www.mikroe.com/forum/

http://www.analog.com/

http://www.microchip.com/

Anexos

Anexo A . Programa Desenvolvido

O "Menu Fabricante" designado no documento, corresponde à rotina menu_escondido e o menu "Pesar" à rotina pesagem, alteração feita aquando da revisão do documento.

```
/*****************************************************************************
        Projecto: Sistema de pesagem para indivíduos com mobilidade diminuída
        Autor: Tiago Andrade @ FEUP 2007/2008
        Disciplina de projecto
*****************************************************************************/
/*
        Objectivos:
                - Aquisição de sinais analógicos de sensores de força
                - Memorizar diferentes taras, utiliza-las e edita-las
                - Fazer corresponder a cada tara um nome e/ou local
                - Pesar com e sem tara

        Descrição do funcionamento:

          Este programa inicia-se com uma pequena apresentação que passados al-
  guns segundos entra num menu constituido por vários submenus e por aí adiante.
           No menu principal pode-se optar por tirar a tara ou pesar. Dentro da
  tara pode-se optar por se faze-la automaticamente, manualmente ou simplesmente
  ver uma tara guardada. Caso de opte pela automática temos a possibilidade de
  manter a monitorização ou reter o valor gravando-o de seguida. Na manual temos
  a possibilidade de introduzir um valor manualmente gravando-o numa memória.
           No submenu de pesagem pode-se optar por uma pesagem total (peso paci-
  ente + tara) com a possibilidade monitorizar o valor do peso, assim como de
  fixa-lo sempre que se pretenda. Caso se opte pela pesagem apenas do paciente é
  necessário escolher a memória correcta da tara, gravada anteriormente, seguin-
  do-se a monitorização com a possibilidade de fixar o valor visualizado.

        Codificação do teclado:

          1   2   3   A   PORTB.F0
          4   5   6   B   PORTB.F1
          7   8   9   C   PORTB.F2
          *   0   #   D   PORTB.F3

           PORTB.F4..7                                                    */

/*****************************************************************************
        Variáveis globais
*****************************************************************************/
unsigned int verificar_apontador,eeprom_hi1,adress_eeprom_lo,adress_eeprom_hi,
ler_tara_lo,ler_tara_hi,ler_tara,peso,posicao,posicao2,posicao_memoria,
posicao_memoria2,adress_eeprom_lo2;
char texto_escrito[14],texto_lido;
short tara_manual_cancel;

/*****************************************************************************
        Rotinas
*****************************************************************************/
void config();
void menu_escondido();
void menu();
void escrita_escondida();
void ler_eeprom2();
void escrita();
void bateria();
void temperatura();
/*****************************************************************************
        Variáveis locais
*****************************************************************************/
short tec;
```

```
59:
60: /*****************************************************************************
61:         Rotina principal
62: *****************************************************************************/
63: void main()
64: {
65:         config();
66:         portc.f0=1;
67:         delay_ms(4000);
68:         portc.f0=0;
69:         bateria();
70:         lcd_custom_cmd(Lcd_CLEAR);
71:         tec=keypad_read();
72:         if (tec==15)
73:               {
74:                   lcd_custom_out(1,1,"solte a tecla");
75:                   while(keypad_read()==15) {}
76:                   tec=keypad_read();
77:                    if (tec==14)
78:                   {
79:                   while(keypad_read()==14) {}
80:                   tec=keypad_read();
81:                   lcd_custom_cmd(Lcd_CLEAR);
82:                    if (tec==1) menu_escondido();
83:                   }
84:                   else return;
85:              }
86:       menu();
87: }
88: /*****************************************************************************
89:         Configurações gerais
90: *****************************************************************************/
91: void config()
92: {
93:         /* Portas */
94:         ADCON1 = 0b00000011;      // Define as entradas analógicas (AN0:AN3)
95: //      ADCON1 = 0b00111001;      // Caso se pretenda utilizar referencias
96:         ADCON2 = 0b00000111; // fRC, 20tAD, left justified
97:         OSCCON = 0b01000111;
98:
99:         TRISA = 0xFF;             // Estabelece as portas A como entradas
100:        TRISB = 0b00011111;       // Estabelece as portas B como entradas e
    saídas
101:        TRISC = 0;                // Estabelece as portas C como saídas
102:        TRISD = 0xFF;             // Estabelece as portas D como entradas
103:        TRISE = 0xFF;             // Estabelece as portas E como entradas
104:
105:        PORTA = 0;                // Inicia a porta A a 0
106:        PORTB = 0;                // Inicia a porta B a 0
107:        PORTC = 0;                // Inicia a porta C a 0
108:        PORTD = 0;                // Inicia a porta D a 0
109:        PORTE = 0;                // Inicia a porta E a 0
110:
111:        keypad_init(&portD);
112:
113:        /* LCD */
114:        Lcd_custom_config(&PORTc,7,6,1,2,&PORTb,7,6,5);         // Inicializa
    LCD ligado à porta B e C
115:        lcd_custom_cmd(Lcd_CLEAR);       // Limpa o ecrã
116:        lcd_custom_cmd(Lcd_CURSOR_OFF);  // Turn cursor off
```

```
117:         lcd_custom_out(1,2,"Balanca Hospitalar");
118:         lcd_custom_out(2,5,"versao 1.0");
119:         lcd_custom_out(3,3,"Tiago F Andrade");
120:         lcd_custom_out(4,3,"FEUP DEMEGI SAIC");
121: }
122:
123:
124: /*****************************************************************************
125:         Rotina le_canal
126:
127:         Descrição: faz o varrimento das portas analógicas definidas obtendo o
128:   valor de cada sensor.
129:
130:         Registo:
131:                   bit 7-6 não implementado: Ler 0.
132:                   bit 5-2 CHS3:CHS0: Selecção dos bits para os diferentes
    canais analógicos
133:                            0000 = Canal 0 (AN0)
134:                            0001 = Canal 1 (AN1)
135:                            0010 = Canal 2 (AN2)
136:                            0011 = Canal 3 (AN3)
137:                            0100 = Canal 4 (AN4)
138:                            0101 = Canal 5 (AN5)
139:                            0110 = Canal 6 (AN6)
140:                            0111 = Canal 7 (AN7)
141:                            1000 = Canal 8 (AN8)
142:                            1001 = Canal 9 (AN9)
143:                            1010 = Canal 10 (AN10)
144:                            1011 = Canal 11 (AN11)
145:                            1100 = Canal 12 (AN12
146:                            1101 = não implementado
147:                            1110 = não implementado
148:                            1111 = não implementado
149:                   bit 1    IR/FEITO: Estado da conversão A/D
150:                            Quando ADON = 1:
151:                            1 = Conversão A/D em progresso
152:                            0 = A/D Inactivo
153:                   bit 0 ADON: bit de habilitação do A/D
154:                            1 = Conversor A/D ligado
155:                            0 = Conversor A/D desligado
156: *****************************************************************************/
157: unsigned int le_canal(char canal)
158: {
159: /*****************************************************************************
160:         Variáveis locais
161: *****************************************************************************/
162: unsigned int adc_lo,adc_hi;
163:
164:         if( canal == 0 ) // Canal 0 (AN0)
165:         {
166:                   ADCON0 = 0b00000011;
167:         }
168:         else if( canal == 1 ) // Canal 1 (AN1)
169:         {
170:                   ADCON0 = 0b00000111;
171:         }
172:         else if( canal == 2 ) // Canal 2 (AN2)
173:         {
174:                   ADCON0 = 0b00001011;
175:         }
```

```
176:         else if( canal == 3 ) // Canal 3 (AN3)
177:         {
178:                 ADCON0 = 0b00001111;
179:         }
180:         else if( canal == 4 ) // Canal 4 (AN4)
181:         {
182:                 ADCON0 = 0b00010011;
183:         }
184:         else if( canal == 5 ) // Canal 5 (AN5)
185:         {
186:                 ADCON0 = 0b00010111;
187:         }
188:         else if( canal == 6 ) // Canal 6 (AN6)
189:         {
190:                 ADCON0 = 0b00011011;
191:         }
192:         else if( canal == 7 ) // Canal 7 (AN7)
193:         {
194:                 ADCON0 = 0b00011111;
195:         }
196:         else if( canal == 8 ) // Canal 8 (AN8)
197:         {
198:                 ADCON0 = 0b00100011;
199:         }
200:         else if( canal == 9 ) // Canal 9 (AN9)
201:         {
202:                 ADCON0 = 0b00100111;
203:         }
204:         else if( canal == 10 ) // Canal 10 (AN10)
205:         {
206:                 ADCON0 = 0b00101011;
207:         }
208:         else if( canal == 11 ) // Canal 11 (AN10)
209:         {
210:                 ADCON0 = 0b00101111;
211:         }
212:
213:         delay_ms(10);
214:         while(ADCON0.GO);        // Espera enquanto faz a conversão
215:         adc_lo = ADRESL >> 4;     // Desloca bits menos significativos
216:         adc_hi = ADRESH << 4;     // Desloca bits mais significativos
217:         return (adc_lo | adc_hi); // Retorna resultados em 12bit [0,4095]
218: }
219: /*************************************************************************
220:         Rotina escreve_canal
221:
222:         Descrição: escreve os valores obtidos em cada um dos canais numa
    variável
223:   associada a cada um dos canais.
224: *************************************************************************/
225: void escreve_canal()
226: {
227: /*************************************************************************
228:         Variáveis locais
229: *************************************************************************/
230:          unsigned int leitura1,leitura2,leitura3,leitura4,leitura5,leitura6,
    leitura7,leitura8,leitura9,leitura10,leitura11,leitura12;
231:          extern unsigned int peso;
232:
233:
```

```
234:
235:                    // sensor 1
236:                    leitura1 = le_canal(0);
237:                    // sensor 2
238:                    leitura2 = le_canal(1);
239:                    // sensor 3
240:                    leitura3 = le_canal(2);
241:                    // sensor 4
242:                    leitura4 = le_canal(3);
243:                    // sensor 5
244:                    leitura5 = le_canal(4);
245:                    // sensor 6
246:                    leitura6 = le_canal(5);
247:                    // sensor 7
248:                    leitura7 = le_canal(6);
249:                    // sensor 8
250:                    leitura8 = le_canal(7);
251:                    // sensor 9
252:                    leitura9 = le_canal(8);
253:                    // sensor 10
254:                    leitura10 = le_canal(9);
255:                    // sensor 11
256:                    leitura11 = le_canal(10);
257:                    // sensor 12
258:                    leitura12 = le_canal(11);
259:
260:            peso = leitura1 + leitura2 + leitura3 + leitura4 + leitura5 + leitura6
    + leitura7 + leitura8 + leitura9 + leitura10 + leitura11 + leitura12;
261:
262: }
263:
264:
265: /*****************************************************************************
266:         Rotina menu
267:
268:         Descrição: dá acesso às rotinas tarar e pesagem.
269: *****************************************************************************/
270:
271: /*****************************************************************************
272:         Rotinas
273: *****************************************************************************/
274: void tarar();
275: void pesagem();
276: void bateria2();
277:
278: /*****************************************************************************
279:         Variáveis locais
280: *****************************************************************************/
281: unsigned short tecla_menu;
282:
283: /*****************************************************************************
284:         Rotina menu
285: *****************************************************************************/
286: void menu()
287: {
288: keypad_init(&portD);
289:
290: while(1)
291: {
292: bateria2();
```

```
293:     lcd_custom_out(1, 8, "Menu");
294:     lcd_custom_out(3, 1, "<-Tarar");
295:     lcd_custom_out(3, 14, "Pesar->");
296:
297: tecla_menu=keypad_read();
298:   if (tecla_menu==14)
299:     {
300:     while(keypad_read()==14){};
301:     lcd_custom_cmd(Lcd_CLEAR);
302:     tarar();
303:     }
304:   if (tecla_menu==16)
305:     {
306:     while(keypad_read()==16){};
307:     lcd_custom_cmd(Lcd_CLEAR);
308:     pesagem();
309:     }
310:   else if (tecla_menu==12)
311:     {
312:     while(keypad_read()==12){};
313:     lcd_custom_cmd(Lcd_CLEAR);
314:     RCON=0b01000000; // ??????????????????
315:     }
316: }
317: lcd_custom_cmd(Lcd_CLEAR);
318: }
319:
320:
321: /*****************************************************************************
322:         Rotina eeprom
323:
324:         Descrição: escreve na memória eeprom.
325: *****************************************************************************/
326:
327: /*****************************************************************************
328:         Biblioteca
329: *****************************************************************************/
330: #include "built_in.h"  // inclusao necessaria para a função Lo() e Hi()
331:
332: /*****************************************************************************
333:         Variáveis globais
334: *****************************************************************************/
335: extern unsigned int adress_eeprom_lo,adress_eeprom_hi,peso,eeprom_hi1,
     verificar_apontador;
336:
337: /*****************************************************************************
338:         Rotinas
339: *****************************************************************************/
340: unsigned int escreve_canal();
341: void escrita();
342:
343: /*****************************************************************************
344:         Variáveis locais
345: *****************************************************************************/
346: unsigned short escreve_eeprom_lo,escreve_eeprom_hi;
347: unsigned int le_eeprom_lo,le_eeprom_hi,le_eeprom;
348: char txt_eeprom[7];
349:
350: /*****************************************************************************
351:         Rotina eeprom
```

```
352: ********************************************************************************/
353: void eeprom()
354: {
355:
356: adress_eeprom_hi=adress_eeprom_lo+240;
357:
358: escreve_eeprom_lo=Lo(peso); //  retorna os 8bits menos significativos
359: escreve_eeprom_hi=Hi(peso); //  retorna os 8bits mais significativos
360:
361: eeprom_write(adress_eeprom_lo,escreve_eeprom_lo);
362: delay_ms(20);
363: eeprom_write(adress_eeprom_hi,escreve_eeprom_hi);
364: delay_ms(20);
365:
366: le_eeprom_lo=eeprom_read(adress_eeprom_lo);
367: le_eeprom_hi=eeprom_read(adress_eeprom_hi);
368:
369: le_eeprom_hi = le_eeprom_hi << 8;  // desloca os algarismos mais significativos
370: le_eeprom = (le_eeprom_lo | le_eeprom_hi);
371: wordtostr(le_eeprom,txt_eeprom); //deve ser para apagar
372: //escrita();
373: if (verificar_apontador!=1) escrita();
374: }
375:
376:
377: /********************************************************************************
378:         Rotina tarar
379:
380:         Descrição: permite memorizar diversas taras através de uma monitorização
381:    ou da introdução de um valor para a tara. Possibilita a visualização da tara
382:    gravada em determinada memória.
383: ********************************************************************************/
384:
385: /********************************************************************************
386:         Variáveis globais
387: ********************************************************************************/
388: extern unsigned int verificar_apontador,eeprom_hi1,adress_eeprom_lo,
     adress_eeprom_lo2,adress_eeprom_hi,ler_tara_lo,ler_tara_hi,ler_tara,peso,
     posicao,posicao2,posicao_memoria,posicao_memoria2;
389: extern char texto_lido, texto_escrito[10];
390: extern short tara_manual_cancel;
391: /********************************************************************************
392:         Rotinas
393: ********************************************************************************/
394: void escreve_canal(), escrita();
395: void verificar_tara(),escrita(),ler_eeprom();
396: void lis_mem(), tara_manual();
397: void bateria2();
398:
399: /********************************************************************************
400:         Variáveis locais
401: ********************************************************************************/
402:
403: unsigned int tecla_tarar1,tecla_tarar2,bandeira_tecla_tara,bandeira_lcd_tara,
     cont_tara,trans_lo,trans_hi;
404: char txt_tarar2[16];
405:
406:
407: /********************************************************************************
408:         Rotina tarar
```

```
409: ********************************************************************************/
410: void tarar()
411: {
412: tecla_tarar1=keypad_read();
413: switch(tecla_tarar1)
414: {
415: case 15: while(keypad_read()==15){};
416: case 0:
417: do
418: {
419: bandeira_tecla_tara=0;
420: tecla_tarar2=keypad_read();
421: bateria2();
422:     lcd_custom_out(1, 5, "Menu Tarar");
423:     lcd_custom_out(3, 1, "<-Auto      Manual->");
424:     lcd_custom_out(4, 5, "V-Ler tara");
425: bandeira_lcd_tara=1;
426:   if (tecla_tarar2==14)
427:     {
428:   cont_tara=499; // tlv seja  necessario ajustar
429:     if (bandeira_lcd_tara==1)
430:             {
431:             lcd_custom_cmd(lcd_clear);
432:             bateria2();
433:             lcd_custom_out(1,1,"   Tara Automatica  ");
434:             lcd_custom_out(3,1,"Tara total:       kg");
435:             lcd_custom_out(4,14,"grave->");
436:             bandeira_lcd_tara=0;
437:             }
438:
439:             while (bandeira_tecla_tara!=16)
440:             {
441:             if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
442:             cont_tara=cont_tara++;
443:             bandeira_tecla_tara=keypad_read();
444:                if (cont_tara==500)
445:                {
446:                   bateria2();
447:                   bandeira_tecla_tara=keypad_read();
448:                   escreve_canal();
449:                   wordToStr(peso,txt_tarar2);
450:                   lcd_custom_out(3,13,txt_tarar2);
451:                   cont_tara=0;
452:                }
453:             }
454:     lcd_custom_cmd(lcd_clear);
455:     bateria2();
456:     lcd_custom_out(1,4,"Memorias livres");
457:     lcd_custom_out(2,2,"1 2 3 4 5 6 7 8 9");
458:     lcd_custom_out(3,1,"Escolha uma memoria");
459:     lcd_custom_out(4,1,"de 1 a 9:");
460:     if (eeprom_read(1) != 0) lcd_custom_out(2,2," "); //else if
     (eeprom_read(21) != 0) lcd_custom_out(2,1," ");
461:     if (eeprom_read(2) != 0) lcd_custom_out(2,4," "); //else lcd_custom_out(2,2,
     " ");
462:     if (eeprom_read(3) != 0) lcd_custom_out(2,6," "); //else lcd_custom_out(2,3,
     " ");
463:     if (eeprom_read(5) != 0) lcd_custom_out(2,8," "); //else lcd_custom_out(2,4,
     " ");
464:     if (eeprom_read(6) != 0) lcd_custom_out(2,10," "); //else lcd_custom_out(2,
     5," ");
```

```
465:     if (eeprom_read(7) != 0) lcd_custom_out(2,12," "); //else lcd_custom_out(2,
    6," ");
466:     if (eeprom_read(9) != 0) lcd_custom_out(2,14," "); //else lcd_custom_out(2,
    7," ");
467:     if (eeprom_read(10) != 0) lcd_custom_out(2,16," "); //else lcd_custom_out(2,
    8," ");
468:     if (eeprom_read(11) != 0) lcd_custom_out(2,18," "); //else if
    (eeprom_read(181) != 0) lcd_custom_out(2,9," ");
469:
470:     eeprom_hi1=0;
471:         while(eeprom_hi1==0 || eeprom_hi1>11 || eeprom_hi1==4 || eeprom_hi1==8)
472:         {
473:         if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
474:         else if (keypad_read()<=11)
475:         {
476:         eeprom_hi1=keypad_read();
477:         adress_eeprom_lo=eeprom_hi1;
478:         }
479:         }
480:     while (keypad_read() != 0){}  //espera que se deixe de premir a tecla
481:     if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
482:     lcd_custom_cmd(Lcd_CLEAR);
483:     verificar_apontador=0;
484:     verificar_tara();
485:     //if (verificar_apontador != 1) {escrita();}
486:
487:     if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
488:
489:     lcd_custom_cmd(Lcd_CLEAR);
490:     }
491:
492:  else if (tecla_tarar2==15)
493:     {
494:     tecla_tarar1=keypad_read();
495:     switch(tecla_tarar1)
496:     {
497:         case 15: while(keypad_read()==15){};
498:         case 0:
499:             lcd_custom_cmd(Lcd_CLEAR);
500:             bateria2();
501:             lcd_custom_out(1, 4, "Lista de taras");
502:             lcd_custom_out(2, 1, "memorizadas");
503:             lcd_custom_out(3, 1, "Escolha uma posicao");
504:             lcd_custom_out(4, 1, "memorizada de 1 a 9");
505:             lis_mem();
506:             lcd_custom_cmd(Lcd_CLEAR);
507:     }
508:     }
509:
510: if (tecla_tarar2==16)
511:     {
512:     tara_manual();
513:     if (tara_manual_cancel==5) return;
514:     else
515:     {
516:             lcd_custom_out(2,10,"       ");
517:             //lcd_custom_out(1,1,"Memorizar tara");
518:             bateria2();
519:             lcd_custom_out(1,4,"memorias livres");
520:             lcd_custom_out(2,2,"1 2 3 4 5 6 7 8 9");
```

```
521:             if (eeprom_read(1) != 0) lcd_custom_out(2,2," "); //else if
     (eeprom_read(21) != 0) lcd_custom_out(2,1," ");
522:             if (eeprom_read(2) != 0) lcd_custom_out(2,4," "); //else
     lcd_custom_out(2,2," ");
523:             if (eeprom_read(3) != 0) lcd_custom_out(2,6," "); //else
     lcd_custom_out(2,3," ");
524:             if (eeprom_read(5) != 0) lcd_custom_out(2,8," "); //else
     lcd_custom_out(2,4," ");
525:             if (eeprom_read(6) != 0) lcd_custom_out(2,10," "); //else
     lcd_custom_out(2,5," ");
526:             if (eeprom_read(7) != 0) lcd_custom_out(2,12," "); //else
     lcd_custom_out(2,6," ");
527:             if (eeprom_read(9) != 0) lcd_custom_out(2,14," "); //else
     lcd_custom_out(2,7," ");
528:             if (eeprom_read(10) != 0) lcd_custom_out(2,16," "); //else
     lcd_custom_out(2,8," ");
529:             if (eeprom_read(11) != 0) lcd_custom_out(2,18," "); //else if
     (eeprom_read(181) != 0) lcd_custom_out(2,9," ");
530:
531:             lcd_custom_out(3,1,"escolha uma memoria");
532:             lcd_custom_out(4,1,"de 1 a 9:");
533:
534:             eeprom_hi1=0;
535:                 while(eeprom_hi1==0 || eeprom_hi1>11 || eeprom_hi1==4 ||
     eeprom_hi1==8)
536:                 {
537:                 if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
538:                 else if (keypad_read()<=11)
539:                 {
540:                 eeprom_hi1=keypad_read();
541:                 adress_eeprom_lo=eeprom_hi1;
542:                 //eeprom_write(13,eeprom_read(eeprom_hi1)); // guarda noutra
     memoria o espaço es
543:                 //eeprom_write(14,eeprom_read(eeprom_hi1+240));
544:                 }
545:                 }
546:     while (keypad_read() != 0){}  //espera que se deixe de premir a tecla
547:     if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
548:     lcd_custom_cmd(Lcd_CLEAR);
549:     verificar_apontador=0;
550:     verificar_tara();            // SUBSTITUIR A VARIAVEL (PESO) NO
     VERIFICAR_TARA E O EEPROM
551:
552:              //if (verificar_apontador==0) escrita();
553:
554:     //if (verificar_apontador=1) {  // repoe a tara guardada anteriormente
555: //                                 eeprom_write(eeprom_hi1,eeprom_read(13));
556: //                                 eeprom_write(eeprom_hi1+240,eeprom_read(14));
557: //                                }
558:
559:
560:
561:     if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
562:
563:     lcd_custom_cmd(Lcd_CLEAR);
564:     }
565:   }
566:
567:   else if (tecla_tarar2==12)
568:     {
```

```
569:     lcd_custom_cmd(Lcd_CLEAR);
570:     return;
571:     }
572: } while (1);
573: }
574: }
575:
576:
577: /*****************************************************************************
578:         Rotina pesagem
579:
580:         Descrição: permite monitorizar o peso total (paciente + tara) ou apenas
581:    o peso do paciente escolhendo-se a tara para a situação em causa.
582: *****************************************************************************/
583:
584: /*****************************************************************************
585:         Biblioteca
586: *****************************************************************************/
587: #include "built_in.h"  // inclusao necessaria para a função Lo() e Hi()
588:
589: /*****************************************************************************
590:         Variáveis globais
591: *****************************************************************************/
592: extern unsigned int eeprom_hi1,adress_eeprom_lo,adress_eeprom_hi,ler_tara_lo,
     ler_tara_hi,ler_tara,peso;
593:
594: /*****************************************************************************
595:         Rotinas
596: *****************************************************************************/
597: void escreve_canal();
598: void eeprom();
599: void lis_mem();
600: void bateria2();
601:
602: /*****************************************************************************
603:         Variáveis locais
604: *****************************************************************************/
605: unsigned int pesagem_lo, pesagem_hi, peso_paciente,peso_paciente2,cont_pesagem,
     voltar,v_1,v_2,v_3,v_4,v_5,cont_variacao,cont_variacao2, vari_anterior,vari_man,
     vari;
606: unsigned short bandeira_lcd_pesagem,bandeira_tecla_pesagem,tecla_pesagem1,
     tecla_pesagem2,bandeira_memoria_pesagem;
607: int variacao,vari_man2;
608: char txt_pesagem[7];
609:
610: /*****************************************************************************
611:         Rotina pesagem
612: *****************************************************************************/
613: void pesagem()
614: {
615: tecla_pesagem1=keypad_read();
616: bandeira_memoria_pesagem=0;
617: switch(tecla_pesagem1)
618: {
619: case 16: while(keypad_read()==16){}; // espera que a pessoa deixe de carregar
620: case 0:
621:
622:
623: do
624: {
```

```
625:   bandeira_tecla_pesagem=0;
626:   tecla_pesagem2=keypad_read();
627:     bateria2();
628:     lcd_custom_out(1, 5, "Menu Pesagem");
629:     lcd_custom_out(3, 1, "<-Total   Paciente->");
630:
631: bandeira_lcd_pesagem=1;
632:   if (tecla_pesagem2==14)
633:     {
634:   cont_pesagem=499; // tlv seja  necessario ajustar
635:            if (bandeira_lcd_pesagem==1)
636:            {
637:            lcd_custom_cmd(Lcd_CLEAR);
638:            bateria2();
639:            lcd_custom_out(1,4,"A monitorizar");
640:            lcd_custom_out(3,1,"Peso total:      kg");
641:            lcd_custom_out(4,14,"reter->");
642:            bandeira_lcd_pesagem=0;
643:            }
644:            while (bandeira_tecla_pesagem!=16 || bandeira_tecla_pesagem==12)
645:            {
646:            cont_pesagem=cont_pesagem++;
647:            bandeira_tecla_pesagem=keypad_read();
648:            if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
649:               if (cont_pesagem==500)
650:               {
651:                  bandeira_tecla_pesagem=keypad_read();
652:                  escreve_canal();
653:                  wordToStr(peso,txt_pesagem);
654:                  lcd_custom_out(3,12,txt_pesagem);
655:                  cont_pesagem=0;
656:                  bateria2();
657:               }
658:            }
659:       lcd_custom_cmd(Lcd_CLEAR);
660:       bateria2();
661:       lcd_custom_out(4,1,"<-voltar");
662:       lcd_custom_out(2,12,txt_pesagem);
663:       lcd_custom_out(2,1,"Peso retido:      kg");
664:           while (keypad_read()!=14)
665:           {
666:             if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
667:             lcd_custom_out(2,13,txt_pesagem);
668:           }
669:     lcd_custom_cmd(Lcd_CLEAR);
670:     }
671:   else if (tecla_pesagem2==16)
672:     {
673:      while (keypad_read()==16){}
674:     lcd_custom_cmd(Lcd_CLEAR);
675:     bateria2();
676:     lcd_custom_out(1, 5, "Peso Paciente   ");
677:     lcd_custom_out(3,1,"ultima configuracao?");
678:     lcd_custom_out(4,1,"<-Sim          Nao->");
679:
680:     do
681:     {
682:           if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);portc.f0=0;return;}
683:           if (keypad_read()==14 && eeprom_read(254)!=0)
684:           {
```

```
685:              variacao=eeprom_read(253);
686:              pesagem_lo=eeprom_read(254);
687:              pesagem_hi=eeprom_read(255);
688:              pesagem_hi = pesagem_hi << 8;  // desloca os algarismos mais
    significativos
689:              ler_tara = (pesagem_lo | pesagem_hi);
690:
691:              bandeira_memoria_pesagem=5;
692:              lcd_custom_cmd(lcd_clear);
693:              bateria2();
694:              wordtostr(variacao,txt_pesagem);
695:              lcd_custom_out(2,1,"Variacao:     kg");
696:              lcd_custom_out(2,9,txt_pesagem);
697:              wordtostr(ler_tara,txt_pesagem);
698:              lcd_custom_out(4,1,"Tara:     kg");
699:              lcd_custom_out(4,5,txt_pesagem);
700:              delay_ms(4000);
701:           }
702:
703:           else if (keypad_read()==16 || eeprom_read(254)==0)
704:           {
705:           bandeira_memoria_pesagem=5;
706:                 lcd_custom_cmd(Lcd_CLEAR);
707:                 bateria2();
708:                 lcd_custom_out(1, 4, "Digite variacao");
709:                 lcd_custom_out(2, 1, "critica:");
710:           lcd_custom_out(4, 19, "->");
711:           lcd_custom_out(3,1,"0kg");
712:           variacao=0;
713:           cont_variacao=1;
714:           cont_variacao2=3;
715:           vari_anterior=1;
716:           v_1=0;
717:           vari=2;
718:           vari_man2=0;
719:
720:                 do{
721:                       vari_anterior=0;
722:                          switch(keypad_read())
723:                          {
724:                 case 1: {while (keypad_read()==1){} if
    (v_1!=1){vari_anterior=1; vari_man=49; vari_man2=1; if
    (vari!=2)cont_variacao++; cont_variacao2++; vari=1; }}break;
725:                 case 2: {while (keypad_read()==2){} if
    (v_1!=1){vari_anterior=1; vari_man=50; vari_man2=2; if
    (vari!=2)cont_variacao++; cont_variacao2++; vari=1; }}break;
726:                 case 3: {while (keypad_read()==3){} if
    (v_1!=1){vari_anterior=1; vari_man=51; vari_man2=3; if
    (vari!=2)cont_variacao++; cont_variacao2++; vari=1; }}break;
727:                 case 4: {while (keypad_read()==4){} if (v_1!=1){if
    (cont_variacao2<10) {cont_variacao++; lcd_custom_chr(3,cont_variacao,'.');
    vari=1; cont_variacao2=10;}}} break;
728:                 case 5: {while (keypad_read()==5){} if
    (v_1!=1){vari_anterior=1; vari_man=52; vari_man2=4; if
    (vari!=2)cont_variacao++; cont_variacao2++; vari=1; }}break;
729:                 case 6: {while (keypad_read()==6){} if
    (v_1!=1){vari_anterior=1; vari_man=53; vari_man2=5; if
    (vari!=2)cont_variacao++; cont_variacao2++; vari=1; }}break;
730:                 case 7: {while (keypad_read()==7){} if
    (v_1!=1){vari_anterior=1; vari_man=54; vari_man2=6; if
    (vari!=2)cont_variacao++; cont_variacao2++; vari=1;}} break;
```

```
731:                case 9: {while (keypad_read()==9){} if
    (v_1!=1){vari_anterior=1; vari_man=55; vari_man2=7; if
    (vari!=2)cont_variacao++; cont_variacao2++; vari=1;}} break;
732:                case 10: {while (keypad_read()==10){} if
    (v_1!=1){vari_anterior=1; vari_man=56; vari_man2=8; if
    (vari!=2)cont_variacao++; cont_variacao2++; vari=1;}} break;
733:                case 11: {while (keypad_read()==11){} if
    (v_1!=1){vari_anterior=1; vari_man=57; vari_man2=9; if
    (vari!=2)cont_variacao++; cont_variacao2++; vari=1;}} break;
734:                case 12: while (keypad_read()==12){}
    lcd_custom_cmd(Lcd_CLEAR);portc.f0=0; return;
735:                case 13: {while (keypad_read()==13){} if
    (v_1!=1){vari_anterior=1; vari_man=48; vari_man2=0; if
    (vari!=2)cont_variacao++; cont_variacao2++; vari=1;}}break;
736:                case 14: {while (keypad_read()==14){} vari_anterior=1; v_1=0;
    cont_variacao=1; vari_man2=0;cont_variacao2=3; lcd_custom_out(3,1," Kg      ");
    vari=2;} break;
737:                case 16: while (keypad_read()==16){} break;
738:                          }
739:                          if (cont_variacao != 0 && vari!=2) {lcd_custom_out(3,
    cont_variacao+1,"kg"); lcd_custom_out(3,cont_variacao+3," ");}
740:
741:                          if (vari_anterior==1) {lcd_custom_chr(3,cont_variacao,
    vari_man); variacao = vari_man2 + variacao*10;}
742:
743:                          if (cont_variacao==1 && vari==2) {lcd_custom_chr(3,1,
    '0'); variacao = 0;}
744:
745:                          if (cont_variacao>=5 || variacao==9999) v_1=1;
    //limita escrita
746:
747:                          if (cont_variacao2==10) v_2=cont_variacao;
748:
749:                        //  wordtostr(variacao,txt_pesagem);
750:                        //  lcd_custom_out(1,1,txt_pesagem);
751:                        eeprom_write(253,variacao); // memorizar variação
752:
753:                      }while (keypad_read()!=16);
754:
755:
756:          while (keypad_read()==16){}
757:
758:          trisc=0x00;
759:          lcd_custom_cmd(Lcd_CLEAR);
760:          bateria2();
761:          lcd_custom_out(1, 4, "Seleccione tara  ");
762:          lcd_custom_out(3, 1, "memorizada de 1 a 9:");
763:          lcd_custom_out(4, 1, "   C para validar   ");
764:
765:
766:          lis_mem();
767:          // memorizar tara escolhida
768:          pesagem_lo=Lo(ler_tara); //  retorna os 8bits menos significativos
769:          pesagem_hi=Hi(ler_tara); //  retorna os 8bits mais significativos
770:          eeprom_write(254,pesagem_lo);
771:          delay_ms(20);
772:          eeprom_write(255,pesagem_hi);
773:          delay_ms(20);
774:
775:          }
```

```
776:     }while(bandeira_memoria_pesagem==0);
777:
778:     bandeira_lcd_pesagem=1;
779:     v_5=1;
780:       while (1)
781:       {
782:       cont_pesagem=499; // tlv seja  necessario ajustar
783:             if (keypad_read() == 14 || bandeira_lcd_pesagem==1)
784:             {
785:             while (keypad_read() == 14){}
786:             lcd_custom_cmd(lcd_clear);
787:             bateria2();
788:             lcd_custom_out(1,4,"A monitorizar:   ");
789:             lcd_custom_out(3,1,"Peso:      kg");
790:             lcd_custom_out(4,14,"reter->");
791:             }
792:             while (keypad_read()!=16)
793:             {
794:             cont_pesagem=cont_pesagem++;
795:             bandeira_tecla_pesagem=keypad_read();
796:             if (keypad_read()==12)
    {lcd_custom_cmd(Lcd_CLEAR);portc.f0=0;return;}
797:             if (keypad_read() == 4) {portc.f0=0; v_5=1;}
798:             if (v_5 == 1) {peso_paciente2=peso-ler_tara; v_5=0;}
799:                if (cont_pesagem==500)
800:                {
801:                   bandeira_tecla_pesagem=keypad_read();
802:                   escreve_canal();
803:                   peso_paciente=peso-ler_tara;
804:                   v_3=peso_paciente+variacao;
805:                   v_4=peso_paciente-variacao;
806:                   if (peso_paciente2<v_4 || peso_paciente2>v_3) portc.f0=1;
807:                   wordToStr(peso_paciente,txt_pesagem);
808:                   lcd_custom_out(3,6,txt_pesagem);
809:                   cont_pesagem=0;
810:                   bateria2();
811:                }
812:             }
813:             lcd_custom_cmd(Lcd_CLEAR);
814:             bateria2();
815:             lcd_custom_out(2,1,"Peso retido:      kg");
816:             lcd_custom_out(4,1,"<-voltar");
817:             lcd_custom_out(2,13,txt_pesagem);
818:          while (keypad_read()!=14)
819:          {
820:            if (keypad_read()==12)
    {lcd_custom_cmd(Lcd_CLEAR);portc.f0=0;return;}
821:             lcd_custom_out(2,13,txt_pesagem);
822:
823:          }
824:     lcd_custom_cmd(Lcd_CLEAR);
825:     }
826:     }
827:
828:   else if (tecla_pesagem2==12)
829:     {
830:     lcd_custom_cmd(Lcd_CLEAR);
831:     return;
832:     }
833: } while (1); ;
```

```
834: }
835: }
836:
837:
838: /*****************************************************************************
839:         Rotina verificar_tara
840:
841:         Descrição: verifica se a memória que se pretende ocupar com determinada
842:   tara já está a ser utilizada ou não deixando a possibilidade do operador subs-
843:   tituíla ou não.
844: *****************************************************************************/
845: /*****************************************************************************
846:         Variáveis globais
847: *****************************************************************************/
848: extern unsigned int adress_eeprom_lo,adress_eeprom_hi,ler_tara_lo,ler_tara_hi,
     ler_tara,verificar_apontador;
849:
850: /*****************************************************************************
851:         Rotinas
852: *****************************************************************************/
853: void eeprom(), escrita();
854: void bateria2();
855: /*****************************************************************************
856:         Variáveis locais
857: *****************************************************************************/
858: unsigned int teste_verificar,adress_verificar,eeprom_verificar;
859: char txt_verificar[7];
860: /*****************************************************************************
861:         Rotina varificar_tara
862: *****************************************************************************/
863: void verificar_tara()
864: {
865: verificar_apontador=0;
866: do
867: {
868:     if (eeprom_read(adress_eeprom_lo) == 0)
869:     {
870:     eeprom();
871:     return;
872:     }
873:     else if (eeprom_read(adress_eeprom_lo) != 0)
874:     {
875:     lcd_custom_cmd(lcd_clear);
876:     bateria2();
877:     lcd_custom_out(1,4,"Tara existente");
878:     lcd_custom_out(2,3,"nessa memoria");
879:     lcd_custom_out(3,2,"deseja alterar?");
880:     lcd_custom_out(4,1,"<-Sim          Nao->");
881:         while (1)
882:         {
883:         if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
884:         teste_verificar=keypad_read();
885:             switch(teste_verificar)
886:             {
887:             case 14: {while (keypad_read() == 14){}verificar_apontador=0;
     eeprom();}
888:             case 16: {while (keypad_read() == 16){}verificar_apontador=1;
     return;}
889:             case 12: {while (keypad_read() == 12){}verificar_apontador=1;
     return;}
```

```
890:             }
891:       }
892:    }
893: }while(1);
894: }
895:
896:
897: /*****************************************************************************
898:         Rotina escrita
899:
900:         Descrição: permite a utilização do teclado disponível com
     caracteristicas
901: aproximadas de um telemovel.
902: *****************************************************************************/
903: /*****************************************************************************
904:         Variáveis globais
905: *****************************************************************************/
906: extern unsigned int posicao,posicao2,posicao_memoria,posicao_memoria2,
     adress_eeprom_lo,adress_eeprom_lo2;
907: extern char texto_escrito[14],texto_lido,texto_lido2;
908:
909: /*****************************************************************************
910:         Rotinas
911: *****************************************************************************/
912: void numerico (kp); // teclas do primeiro andar teclado
913: void alfa1 (kp); // teclas do segundo andar do teclado
914: void alfa2 (kp); // teclas do terceiro andar do teclado
915: void alfa3 (kp); // teclas do quarto andar do teclado
916: void ler_eeprom(),escrever_eeprom();
917:
918: /*****************************************************************************
919:         Variáveis locais
920: *****************************************************************************/
921: unsigned short kp, memkp, cnt, oldstate = 0, ola;
922: long ola2;
923: unsigned int cont_escrita;
924: //char bolas[16];
925:
926: /*****************************************************************************
927:         Rotina escrita
928: *****************************************************************************/
929: void escrita()
930: {
931:
932: lcd_custom_cmd(lcd_clear);
933: lcd_custom_cmd(LCD_CURSOR_OFF);
934: lcd_custom_out(2,1,"escreva");              // config_teclado();
935: posicao=0;
936: posicao2=1;
937: ola=0;
938: memkp = 0;
939: cont_escrita=0;
940: ola2=600;
941:
942:   do {
943:
944:     kp = 0;
945:
946:     if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
947:     do {
```

```
948:       kp = Keypad_Read();
949:     if (kp == 0) {ola2++;/*longtostr(ola2,bolas);lcd_custom_out(2,1,bolas);*/ }
950:
951:                                     if (kp == 15)
952:                                     {
953:                                     while(keypad_read()==15){}
954:                                     if (keypad_read()==12)
     {lcd_custom_cmd(Lcd_CLEAR);return;}
955:                                     lcd_custom_cmd(LCD_CLEAR);
956:
957:                                            while (posicao2<15) // acaba de
     preencher o espaço
958:                                            {
959:                                            posicao2++;
960:                                            texto_escrito[posicao2]=160;
961:                                            }
962:                                     escrever_eeprom();
963:                                     ler_eeprom();
964:                                        return;
965:                                     }
966:                                     else if (keypad_read()==12)
     {lcd_custom_cmd(Lcd_CLEAR);return;}
967:       }    while (!kp);
968:
969:       if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
970:       if (memkp != kp) {cnt=1;}
971:       if (ola!=0 && memkp != kp) {posicao2++;} // avança ao mudar de tecla
972:       if (ola2>400 && memkp == kp) {posicao2++; cnt=1;} // avança ao esperar
973:       memkp = kp;
974:
975:       ola++;
976:     switch (cnt) {
977:       case  1: numerico (); break;
978:       case  2: alfa1 (); break;
979:       case  3: alfa2 (); break;
980:       case  4: alfa3 (); break;
981:     if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
982:     }
983:     // Prepare value for output
984:     if (kp != oldstate && kp == cnt) {
985:       oldstate = kp;
986:       while(!Keypad_Released())
987:         asm nop;
988:       }
989:     else {
990:       cnt++;
991:       while(!Keypad_Released())
992:         asm nop;
993:       }
994:      if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
995:     // Escrita no LCD
996:        if (posicao2<=14)
997:        {
998:        if (keypad_read()==12) {lcd_custom_cmd(Lcd_CLEAR);return;}
999:        lcd_custom_chr(1, posicao2, kp);
1000:       texto_escrito[posicao2]=kp;
1001:       lcd_custom_out(2,1,"                ");
1002:       lcd_custom_out(3,1,"                ");
1003:       lcd_custom_out(4,6,"V-grave");
1004:            if (posicao2 == 14) {
```

```
1005:                                lcd_custom_out(3,1,"ultimo caracter");
1006:                               }
1007:        }
1008:        if (posicao2>14) {
1009:
1010:                               lcd_custom_cmd(LCD_CURSOR_OFF);
1011:                               lcd_custom_out(3,1,"                    ");
1012:                               lcd_custom_out(3,1,"ocupou escrita");
1013:                               posicao2 = 14;
1014:                            }
1015:
1016:        if (cnt == 5) cnt = 1;
1017:
1018:        else if (kp==16 || kp==14) cnt=1;
1019:
1020:
1021:    } while (1);
1022: }
1023:
1024: /*****************************************************************************
1025:         Rotina numerico
1026: *****************************************************************************/
1027: void numerico (kp){
1028: switch (kp) {
1029:
1030:       case  1: kp = 49; ola2=0; break; // 1
1031:       case  2: kp = 50; ola2=0; break; // 2
1032:       case  3: kp = 51; ola2=0; break; // 3
1033:       case  4: kp = 45; ola2=0; break; // -
1034:       case  5: kp = 52; ola2=0; break; // 4
1035:       case  6: kp = 53; ola2=0; break; // 5
1036:       case  7: kp = 54; ola2=0; break; // 6
1037:       case  8: kp = 46; ola2=0; break; // .
1038:       case  9: kp = 55; ola2=0; break; // 7
1039:       case 10: kp = 56; ola2=0; break; // 8
1040:       case 11: kp = 57; ola2=0; break; // 9
1041: //       case 12: tarar(); // C
1042:       case 13: kp = 48; break; // 0
1043:       case 14: if (posicao2 == 1) posicao2=0; else if (posicao2 > 1){kp = 160;
      posicao2--;ola=0;ola2=0;} break;
1044:       case 15: break;
1045:       case 16: kp = 160; break;
1046:
1047:     }
1048: }
1049:
1050: /*****************************************************************************
1051:         Rotina alfa1
1052: *****************************************************************************/
1053: void alfa1 (kp){
1054: switch (kp) {
1055:
1056:       case 1: kp = 65; ola2=0; break; // A
1057:       case 2: kp = 68; ola2=0; break; // D
1058:       case 3: kp = 71; ola2=0; break; // G
1059:       case 4: kp = 42; ola2=0; break; // *
1060:       case 5: kp = 74; ola2=0; break; // J
1061:       case 6: kp = 77; ola2=0; break; // M
1062:       case 7: kp = 80; ola2=0; break; // P
1063:       case 8: kp = 44; ola2=0; break; // ,
```

```
1064:       case 9: kp = 83; ola2=0; break; // S
1065:       case 10: kp = 86; ola2=0; break; // V
1066:       case 11: kp = 89; ola2=0; break; // Y
1067: //      case 12: return; break; //
1068:       case 13: kp = 63; break; // ?
1069:       case 14: if (posicao2 == 1) posicao2=0; else if (posicao2 > 1){kp = 160;
      posicao2--;ola=0;ola2=0;} break;
1070:       case 15: break;
1071:       case 16: kp = 160; break;
1072:     }
1073: }
1074:
1075: /*****************************************************************************
1076:         Rotina alfa2
1077: *****************************************************************************/
1078: void alfa2 (kp){
1079: switch (kp) {
1080:
1081:       case 1: kp = 66; ola2=0; break; // B
1082:       case 2: kp = 69; ola2=0; break; // E
1083:       case 3: kp = 72; ola2=0; break; // H
1084:       case 4: kp = 38; ola2=0; break; // _
1085:       case 5: kp = 75; ola2=0; break; // K
1086:       case 6: kp = 78; ola2=0; break; // N
1087:       case 7: kp = 81; ola2=0; break; // Q
1088:       case 8: kp = 58; ola2=0; break; // :
1089:       case 9: kp = 84; ola2=0; break; // T
1090:       case 10: kp = 87; ola2=0; break; // W
1091:       case 11: kp = 90; ola2=0; break; // Z
1092: //      case 12: return; break; //
1093:       case 13: kp = 40; break; // (
1094:       case 14: if (posicao2 == 1) posicao2=0; else if (posicao2 > 1){kp = 160;
      posicao2--;ola=0;ola2=0;} break;
1095:       case 15: break;
1096:       case 16: kp = 160; break;
1097:     }
1098: }
1099:
1100: /*****************************************************************************
1101:         Rotina alfa3
1102: *****************************************************************************/
1103: void alfa3 (kp){
1104: switch (kp) {
1105:
1106:       case 1: kp = 67; ola2=0; break; // C
1107:       case 2: kp = 70; ola2=0; break; // F
1108:       case 3: kp = 73; ola2=0; break; // I
1109:       case 4: kp = 43; ola2=0; break; // +
1110:       case 5: kp = 76; ola2=0; break; // L
1111:       case 6: kp = 79; ola2=0; break; // O
1112:       case 7: kp = 82; ola2=0; break; // R
1113:       case 8: kp = 59; ola2=0; break; // ;
1114:       case 9: kp = 85; ola2=0; break; // U
1115:       case 10: kp = 88; ola2=0; break; // X
1116:       case 11: kp = 199; ola2=0; break; // ç
1117: //      case 12: return; break; //
1118:       case 13: kp = 41; break; // )
1119:       case 14: if (posicao2 == 1) posicao2=0; else if (posicao2 > 1){kp = 160;
      posicao2--;ola=0;ola2=0;} break;
1120:       case 15: break;
```

```
1121:       case 16: kp = 160; break;
1122:
1123:     }
1124: }
1125:
1126:
1127: /*************************************************************************
1128:         Rotina ler_eeprom
1129:
1130:         Descrição: ciclo for para leitura dos dados guardados na memoria eeprom.
1131: *************************************************************************/
1132: /*************************************************************************
1133:         Variáveis globais
1134: *************************************************************************/
1135: extern char texto_lido;
1136: extern unsigned int posicao,posicao2,eeprom_hi1;
1137:
1138: /*************************************************************************
1139:         Variáveis locais
1140: *************************************************************************/
1141: unsigned int ler_ee;
1142:
1143: /*************************************************************************
1144:         Rotina ler_eeprom
1145: *************************************************************************/
1146: void ler_eeprom()
1147: {
1148:      posicao2=6;
1149:      posicao=0;
1150:      ler_ee=eeprom_hi1*20;
1151:      for (posicao = ler_ee; posicao < ler_ee+15u; posicao++)
1152:      {
1153:          texto_lido = Eeprom_Read(posicao);
1154:          lcd_custom_chr(2,posicao2++,texto_lido);
1155:          lcd_custom_out(2,1,"Nome: ");
1156:      }
1157:       return;
1158: }
1159:
1160:
1161: /*************************************************************************
1162:         Rotina escrever_eeprom
1163:
1164:         Descrição: ciclo for para escrita dos dados na memoria eeprom.
1165: *************************************************************************/
1166: /*************************************************************************
1167:         Variáveis globais
1168: *************************************************************************/
1169: extern char texto_escrito[14];
1170: extern unsigned int posicao,posicao2,eeprom_hi1;
1171:
1172: /*************************************************************************
1173:         Variáveis rotinas
1174: *************************************************************************/
1175: void ler_eeprom();
1176:
1177: /*************************************************************************
1178:         Variáveis locais
1179: *************************************************************************/
1180: unsigned int esc_ee;
```

```
1181:
1182: /*************************************************************************
1183:         Rotina escrever_eeprom
1184: *************************************************************************/
1185: void escrever_eeprom()
1186: {
1187:      posicao2=0;
1188:      posicao=0;
1189:      esc_ee=eeprom_hi1*20;
1190:
1191:      for (posicao = esc_ee; posicao < esc_ee+15u; posicao++)
1192:           EEprom_Write(posicao, texto_escrito[posicao2++]);
1193:
1194: return;
1195: }
1196:
1197:
1198: /*************************************************************************
1199:         Rotina menu_escondido
1200:
1201:         Descrição: dá acesso às rotinas apagar_eeprom e calibrar.
1202: *************************************************************************/
1203: /*************************************************************************
1204:         Rotinas
1205: *************************************************************************/
1206: void apagar_eeprom();
1207: void calibrar();
1208: void menu();
1209: /*************************************************************************
1210:         Variáveis locais
1211: *************************************************************************/
1212: unsigned short tecla_menu_escondido;
1213:
1214: /*************************************************************************
1215:         Rotina menu_escondido
1216: *************************************************************************/
1217: void menu_escondido()
1218: {
1219: keypad_init(&portd);
1220: lcd_custom_cmd(lcd_clear);
1221: while(1)
1222: {
1223:     if (portb.f0 == 0) lcd_custom_out(4,3,"Baterias Fracas");
1224:     lcd_custom_out(1, 2, "Menu do Fabricante");
1225:     lcd_custom_out(3, 1, "EEPROM    Calibracao");
1226:
1227: tecla_menu_escondido=keypad_read();
1228:   if (tecla_menu_escondido==14)
1229:     {
1230:     while(keypad_read()==14){};
1231:     lcd_custom_cmd(Lcd_CLEAR);
1232:     apagar_eeprom();
1233:     }
1234:   if (tecla_menu_escondido==16)
1235:     {
1236:     while(keypad_read()==16){};
1237:     lcd_custom_cmd(Lcd_CLEAR);
1238:     calibrar();
1239:     }
1240:   else if (tecla_menu_escondido==12)
```

```
1241:     {
1242:     while(keypad_read()==12){};
1243:     lcd_custom_cmd(Lcd_CLEAR);
1244:     menu();
1245:     }
1246: }
1247: lcd_custom_cmd(Lcd_CLEAR);
1248: }
1249:
1250:
1251: /*****************************************************************************
1252:         Rotina apagar_eeprom
1253:
1254:         Descrição: permite levar a zero todos os espaços da memória eeprom, é
1255: necessário ser executada no início da primeira utilização do equipamento.
1256: *****************************************************************************/
1257: /*****************************************************************************
1258:         Variáveis globais
1259: *****************************************************************************/
1260: extern unsigned int posicao,posicao2,eeprom_hi1;
1261:
1262: /*****************************************************************************
1263:         Rotinas
1264: *****************************************************************************/
1265: void bateria2();
1266: /*****************************************************************************
1267:         Rotina apagar_eeprom
1268: *****************************************************************************/
1269: void apagar_eeprom()
1270: {
1271: bateria2();
1272:     lcd_custom_out(1,5,"Menu EEPROM     ");
1273:     lcd_custom_out(3,1,"deseja mesmo apagar?");
1274:     lcd_custom_out(4,1,"<-Nao          Sim->");
1275:         while (1)
1276:         {
1277:             switch(keypad_read())
1278:             {
1279:             case 16: {
1280:                         lcd_custom_cmd(lcd_clear);
1281:                         bateria2();
1282:                         lcd_custom_out(2,2,"Apagando memoria");
1283:                         while (keypad_read() == 14){}
1284:                         for (posicao = 0; posicao < 256u; posicao++) // coloca
    eeprom com zeros
1285:                              EEprom_Write(posicao, 0);
1286:                         delay_ms(20);
1287:
1288:                         for (posicao = 20; posicao < 239u; posicao++) // coloca
    eeprom com espaços na memoria escrita
1289:                              EEprom_Write(posicao, 160);
1290:
1291:
1292:                         lcd_custom_cmd(lcd_clear);
1293:                         bateria2();
1294:                         lcd_custom_out(2,4,"Memoria EEprom");
1295:                         lcd_custom_out(4,7,"Apagada");
1296:                         delay_ms(2000);
1297:                         lcd_custom_cmd(lcd_clear);
1298:                         return;
```

```
1299:                       }
1300:              case 14: {while (keypad_read() == 14){} lcd_custom_cmd(lcd_clear);
     return;}
1301:              case 12: {while (keypad_read() == 12){} lcd_custom_cmd(lcd_clear);
     return;}
1302:              }
1303:       }
1304:     lcd_custom_cmd(lcd_clear);
1305: }
1306:
1307:
1308: /*****************************************************************************
1309:         Rotina calibrar
1310:
1311:         Descrição: calibra os sensores utilizados.
1312: *****************************************************************************/
1313: /*****************************************************************************
1314:         Variáveis globais
1315: *****************************************************************************/
1316:
1317:
1318: void calibrar()
1319: {
1320:
1321:
1322: }
1323:
1324:
1325: /*****************************************************************************
1326:         Rotina lis_mem
1327:
1328:         Descrição: permite visualizar os dados contidos em cada uma das nove me-
1329: mórias existentes.
1330: *****************************************************************************/
1331: /*****************************************************************************
1332:         Variáveis globais
1333: *****************************************************************************/
1334: extern unsigned int verificar_apontador, eeprom_hi1,adress_eeprom_lo,
     adress_eeprom_lo2,adress_eeprom_hi,ler_tara_lo,ler_tara_hi,ler_tara,peso,
     posicao,posicao2,posicao_memoria,posicao_memoria2;
1335:
1336: /*****************************************************************************
1337:         Rotinas
1338: *****************************************************************************/
1339: void ler_eeprom(), escrita(), eeprom(), escreve_canal(), tara_manual();
1340:
1341: /*****************************************************************************
1342:         Variáveis locais
1343: *****************************************************************************/
1344: char txt_lis_mem[7], txt_lis_mem2, txt_lis_mem3, l1, l2, l3;
1345: short lis=0,lis2,lis3;
1346: unsigned int lis_mem_anterior, lis_mem_man, cont_lis_mem, bandeira_lcd_lis_mem,
     bandeira_tecla_lis_mem, cont_lis_mem2, tar_lis_mem, cont_lis_mem3, lis_mem_1,
     lis_mem_2,lis_mem_3,lis_mem_4,lis_mem_5,lis_mem_ciclo;
1347: float valor_lis_mem, lis_mem_man2;
1348:
1349:
1350: void lis_mem()
1351: {
1352: while (1)
```

```
1353: {
1354:             eeprom_hi1=0;
1355:                 while(eeprom_hi1==0)
1356:                 {
1357:                 if (keypad_read()==12) {while (keypad_read()==12){}
      lcd_custom_cmd(Lcd_CLEAR);return;}
1358:                 eeprom_hi1=keypad_read();
1359:                 if (lis3 == 1) eeprom_hi1=1;
1360:                 adress_eeprom_lo=eeprom_hi1;
1361:                 adress_eeprom_hi=adress_eeprom_lo+240;
1362:                 }
1363: lis=0;
1364: lcd_custom_cmd(lcd_clear);
1365:    switch (eeprom_hi1){
1366:       case 1: txt_lis_mem2=49; lis2=1; lis3=0; break;
1367:       case 2: txt_lis_mem2=50; lis2=2; lis3=0; break;
1368:       case 3: txt_lis_mem2=51; lis2=3; lis3=0; break;
1369:       case 4: lcd_custom_cmd(lcd_clear); lcd_custom_out(1,1,"seleccao errada");
      delay_ms(1000); lcd_custom_cmd(lcd_clear); lis3=1; break;
1370:       case 5: txt_lis_mem2=52; lis2=5; break;
1371:       case 6: txt_lis_mem2=53; lis2=6; break;
1372:       case 7: txt_lis_mem2=54; lis2=7; break;
1373:       case 8: lcd_custom_cmd(lcd_clear); lcd_custom_out(1,1,"seleccao errada");
      delay_ms(1000); lcd_custom_cmd(lcd_clear); lis3=1; break;
1374:       case 9: txt_lis_mem2=55; lis2=9; break;
1375:       case 10: txt_lis_mem2=56; lis2=10; break;
1376:       case 11: txt_lis_mem2=57; lis2=11; break;
1377:       case 12: while(keypad_read()==12){} lcd_custom_cmd(lcd_clear); return;
1378:       case 13: lcd_custom_cmd(lcd_clear); lcd_custom_out(1,1,"seleccao
      errada"); delay_ms(1000); lcd_custom_cmd(lcd_clear); lis3=1; break;
1379:       case 14: lcd_custom_cmd(lcd_clear); lcd_custom_out(1,1,"seleccao
      errada"); delay_ms(1000); lcd_custom_cmd(lcd_clear); lis3=1; break;
1380:       case 15: lis=1; while (keypad_read()==15){} break;
1381:       case 16: lcd_custom_cmd(lcd_clear); lcd_custom_out(1,1,"seleccao
      errada"); delay_ms(1000); lcd_custom_cmd(lcd_clear); lis3=1; break;
1382:    }
1383:        if (lis != 1)
1384:        {
1385:             ler_tara_lo=eeprom_read(adress_eeprom_lo);
1386:             ler_tara_hi=eeprom_read(adress_eeprom_hi);
1387:             ler_tara_hi = ler_tara_hi << 8;  // desloca os algarismos mais
      significativos
1388:             ler_tara = (ler_tara_lo | ler_tara_hi);
1389:             wordtostr(ler_tara,txt_lis_mem);
1390:
1391:             if (keypad_read()==12) {while (keypad_read()==12){}
      lcd_custom_cmd(Lcd_CLEAR);return;}
1392:             lcd_custom_out(1,1,"Memoria");
1393:             lcd_custom_chr(1,9,txt_lis_mem2);
1394:             lcd_custom_out(1,19,"kg");
1395:             lcd_custom_out(1,13,txt_lis_mem);
1396:             ler_eeprom();
1397:             lcd_custom_out(4,3,"V-Editar campos");
1398:        }
1399:        while (lis==1)
1400:        {
1401:             eeprom_hi1=lis2;
1402:             adress_eeprom_lo=eeprom_hi1;
1403:             adress_eeprom_hi=adress_eeprom_lo+240;
1404:             ler_tara_lo=eeprom_read(adress_eeprom_lo);
```

```
1405:             ler_tara_hi=eeprom_read(adress_eeprom_hi);
1406:             ler_tara_hi = ler_tara_hi << 8;  // desloca os algarismos mais
      significativos
1407:             ler_tara = (ler_tara_lo | ler_tara_hi);
1408:             wordtostr(ler_tara,txt_lis_mem);
1409:             lcd_custom_out(1,1,"Editar?");
1410:             lcd_custom_out(1,15,"kg");
1411:             lcd_custom_out(1,9,txt_lis_mem);
1412:             lcd_custom_out(3,1,"<-Sim          Nao->");
1413:             if (keypad_read()==12) {while (keypad_read()==12){}
      lcd_custom_cmd(Lcd_CLEAR);return;}
1414:             if (keypad_read()==16) {while (keypad_read()==16){}
      lcd_custom_cmd(Lcd_CLEAR);return;}
1415:             if (keypad_read()==14)
1416:             {
1417:             do
1418:             {
1419:               while (keypad_read()==14){}
1420:               lcd_custom_out(1,1,"Tara            Nome");
1421:               lcd_custom_out(3,1,"<-                ->");
1422:                           do
1423:                           {
1424:                               if (keypad_read()==14)
1425:                               {
1426:                               while (keypad_read()==14){}
1427:                               lcd_custom_out(1,1,"       Tara         ");
1428:                               lcd_custom_out(3,1,"<-Auto      Manual->");
1429:                                     while(1)
1430:                                     {
1431:                                         if (keypad_read()==14)
1432:                                         {
1433:                                         while (keypad_read()==14){}
1434:                                         lcd_custom_cmd(lcd_clear);
1435:                                         bandeira_tecla_lis_mem=keypad_read();
1436:                                           cont_lis_mem=49; // tlv seja
      necessario ajustar
1437:                                           if (bandeira_lcd_lis_mem==1)
1438:                                                   {
1439:                                                   lcd_custom_cmd(Lcd_CLEAR);
1440:                                                   bandeira_lcd_lis_mem=0;
1441:                                                   }
1442:
1443:                                                   while
      (bandeira_tecla_lis_mem!=16)
1444:                                                   {
1445:                                                   lcd_custom_out(1,1,"Tara
      total:");
1446:                                                   lcd_custom_out(3,7,"kg");
1447:                                                   lcd_custom_out(4,14,"grave-
      >");
1448:                                                   if (keypad_read()==12)
      {lcd_custom_cmd(Lcd_CLEAR);return;}
1449:                                                   cont_lis_mem=cont_lis_mem++;
1450:
      bandeira_tecla_lis_mem=keypad_read();
1451:                                                      if (cont_lis_mem==50)
1452:                                                      {
1453:
      bandeira_tecla_lis_mem=keypad_read();
1454:                                                         escreve_canal();
```

```
1455:                                                           wordtostr(peso,
     txt_lis_mem3);
1456:                                                           lcd_custom_out(3,1,
     txt_lis_mem3);
1457:                                                           cont_lis_mem=0;
1458:                                                       }
1459:                                                   }
1460:                                               while (keypad_read()==16){}
1461:                                               eeprom();
1462:                                           if (keypad_read()==12)
     {lcd_custom_cmd(Lcd_CLEAR);return;}
1463:
1464:                                           lcd_custom_cmd(Lcd_CLEAR);
1465:                                           return;
1466:                                         }
1467:
1468:
1469:
1470:
1471:
1472:
1473:                                         if (keypad_read()==16)
1474:                                         {
1475:                                         while (keypad_read()==16){}
1476:                                         tara_manual();
1477:                                         eeprom();
1478:                                           if (keypad_read()==12)
     {lcd_custom_cmd(Lcd_CLEAR);return;}
1479:
1480:                                           lcd_custom_cmd(Lcd_CLEAR);
1481:                                           return;
1482:                   }
1483:
1484:
1485:
1486:
1487:
1488:
1489:
1490:                                       if (keypad_read()==12) {while
     (keypad_read()==12){} lcd_custom_cmd(Lcd_CLEAR);return;}
1491:                                       }
1492:                                 }
1493:                                 if (keypad_read()==16)
1494:                                 {
1495:                                 while (keypad_read()==16){}
1496:                                 lcd_custom_cmd(lcd_clear);
1497:                                 ler_eeprom();
1498:                                 lcd_custom_out(1,1,"Dado a alterar:");
1499:                                 delay_ms(1500);
1500:                                 escrita();
1501:                                 lcd_custom_cmd(lcd_clear);
1502:                                 return;
1503:                                 }
1504:                               if (keypad_read()==12) {while (keypad_read()==12){}
     lcd_custom_cmd(Lcd_CLEAR);return;}
1505:                                 }while(1);
1506:             if (keypad_read()==12) {while (keypad_read()==12){}
     lcd_custom_cmd(Lcd_CLEAR);return;}
1507:             } while(1);
```

```
1508:             }
1509:         }
1510: }
1511: }
1512:
1513:
1514: /*****************************************************************************
1515:         Rotina tara_manual
1516:
1517:         Descrição: permite a incerção de um valor de Tara de forma manual.
1518: *****************************************************************************/
1519: /*****************************************************************************
1520:         Variáveis globais
1521: *****************************************************************************/
1522: extern unsigned int peso;
1523: extern short tara_manual_cancel;
1524:
1525: /*****************************************************************************
1526:         Rotinas
1527: *****************************************************************************/
1528: void bateria2();
1529:
1530: /*****************************************************************************
1531:         Variáveis locais
1532: *****************************************************************************/
1533: unsigned int cont_tara_man, cont_tara_man2, cont_tara_man3, tara_man,tar, tar2,
     tara_anterior;
1534: unsigned int t_1,t_2,t_3,t_4,t_5,t_ciclo;
1535: char txt_tarar[16],t1,t2,t3;
1536: float valor_tara_m, tara_man2;
1537:
1538:
1539: void tara_manual()
1540: {
1541:     lcd_custom_cmd(lcd_clear);
1542:     bateria2();
1543:     lcd_custom_out(1, 4, "Tara Manual");
1544:    // lcd_custom_out(1,1,"Tara manual:");   //INTRODUZIR TARA MANUAL
1545:     lcd_custom_out(3,1,"0Kg");   // repetido em baixo???????????
1546:     lcd_custom_out(4,14,"grave->");
1547:     cont_tara_man=0;
1548:     valor_tara_m=0;
1549:     tar=1;
1550:     //lcd_custom_out(2,1,"0Kg      ");
1551:     cont_tara_man2=3;
1552:     t_1=0;
1553:     t_2=0;
1554:     t_3=0;
1555:     tara_manual_cancel=0;
1556:     do{
1557:        tara_anterior=0;
1558:           switch(keypad_read())
1559:           {
1560: case 1: {while (keypad_read()==1){} if (t_1!=1){tara_anterior=1; tara_man=49;
      tara_man2=1; if (tar!=2)cont_tara_man++; cont_tara_man2++; tar=1; }}break;
1561: case 2: {while (keypad_read()==2){} if (t_1!=1){tara_anterior=1; tara_man=50;
      tara_man2=2; if (tar!=2)cont_tara_man++; cont_tara_man2++; tar=1; }}break;
1562: case 3: {while (keypad_read()==3){} if (t_1!=1){tara_anterior=1; tara_man=51;
      tara_man2=3; if (tar!=2)cont_tara_man++; cont_tara_man2++; tar=1; }}break;
1563: case 4: {while (keypad_read()==4){} if (t_1!=1){if (cont_tara_man2<10)
      {cont_tara_man++; lcd_custom_chr(3,cont_tara_man,'.'); tar=1;
      cont_tara_man2=10;}}} break;
```

```
1564: case 5: {while (keypad_read()==5){} if (t_1!=1){tara_anterior=1; tara_man=52;
      tara_man2=4; if (tar!=2)cont_tara_man++; cont_tara_man2++; tar=1; }}break;
1565: case 6: {while (keypad_read()==6){} if (t_1!=1){tara_anterior=1; tara_man=53;
      tara_man2=5; if (tar!=2)cont_tara_man++; cont_tara_man2++; tar=1; }}break;
1566: case 7: {while (keypad_read()==7){} if (t_1!=1){tara_anterior=1; tara_man=54;
      tara_man2=6; if (tar!=2)cont_tara_man++; cont_tara_man2++; tar=1;}} break;
1567: case 9: {while (keypad_read()==9){} if (t_1!=1){tara_anterior=1; tara_man=55;
      tara_man2=7; if (tar!=2)cont_tara_man++; cont_tara_man2++; tar=1;}} break;
1568: case 10: {while (keypad_read()==10){} if (t_1!=1){tara_anterior=1; tara_man=56;
      tara_man2=8; if (tar!=2)cont_tara_man++; cont_tara_man2++; tar=1;}} break;
1569: case 11: {while (keypad_read()==11){} if (t_1!=1){tara_anterior=1; tara_man=57;
      tara_man2=9; if (tar!=2)cont_tara_man++; cont_tara_man2++; tar=1;}} break;
1570: case 12: while (keypad_read()==12){} tara_manual_cancel=5;
      lcd_custom_cmd(Lcd_CLEAR); return;
1571: case 13: {while (keypad_read()==13){} if (t_1!=1){tara_anterior=1; tara_man=48;
      tara_man2=0; if (tar!=2)cont_tara_man++; cont_tara_man2++; tar=1;}}break;
1572: case 14: {while (keypad_read()==14){} tara_anterior=1; t_1=0; cont_tara_man=1;
      tara_man2=0;cont_tara_man2=3; lcd_custom_out(2,1," Kg      "); tar=2;} break;
1573: case 16: while (keypad_read()==16){} break;
1574:           }
1575:           if (cont_tara_man != 0 && tar!=2) {lcd_custom_out(3,cont_tara_man+1,
      "Kg"); lcd_custom_out(3,cont_tara_man+3," ");}
1576:
1577:           if (tara_anterior==1) {lcd_custom_chr(3,cont_tara_man,tara_man);
      valor_tara_m = tara_man2 + valor_tara_m*10;}
1578:
1579:           if (cont_tara_man==1 && tar==2) {lcd_custom_chr(3,1,'0');
      valor_tara_m = 0;}
1580:
1581:           if (cont_tara_man>=5 || valor_tara_m==9999) t_1=1; //limita escrita
1582:
1583:           if (cont_tara_man2==10) {t_2=cont_tara_man; t_3=1;} // número de
      valores depois da virgula
1584:           else if (t_3==0) t_2=cont_tara_man;
1585:
1586:  floattostr(valor_tara_m,txt_tarar);
1587:           lcd_custom_out(1,1,txt_tarar);
1588:
1589:
1590:
1591:       }while (keypad_read()!=16);
1592:           t_2 = cont_tara_man - t_2;
1593:                       wordtostr(t_2,t1);
1594:                       lcd_custom_out(1,8,t1);
1595:                       delay_ms(2000);
1596:         //t_4=pow(10, t_2); //expoente  ocupa 17%
1597:         if (t_2==0) t_4=1;
1598:         if (t_2==1) t_4=10;
1599:         if (t_2==2) t_4=100;
1600:         if (t_2==3) t_4=1000;
1601:         if (t_2==4) t_4=10000;
1602:                       wordtostr(t_4,t2);
1603:                       lcd_custom_out(1,8,t2);
1604:                       delay_ms(2000);
1605:           valor_tara_m = valor_tara_m / t_4;
1606:           floattostr(valor_tara_m,txt_tarar);
1607:           lcd_custom_out(1,1,txt_tarar);
1608:           delay_ms(2000);
1609:           peso = valor_tara_m;
1610:           wordtostr(peso,txt_tarar);
```

```
1611:           lcd_custom_cmd(lcd_clear);
1612:           lcd_custom_out(1,1,txt_tarar);
1613:           delay_ms(4000);
1614:
1615: }
1616:
1617:
1618: /*****************************************************************************
1619:         Rotina bateria
1620:
1621:         Descrição: verifica se o nivel lógico da porte.c3 correspondes
1622: a 1 (se a bateria tiver tensão) ou a 0 (se a bateria tiver uma tensão abaixo da
1623: estipulada).
1624: *****************************************************************************/
1625:
1626: /*****************************************************************************
1627:         Rotina bateria
1628: *****************************************************************************/
1629: void bateria()
1630: {
1631:       if (portb.f0 == 0)
1632:       {
1633:          lcd_custom_cmd(lcd_clear);
1634:          lcd_custom_out(1,1,"Baterias Fracas (BF)");
1635:          lcd_custom_out(3,5,"Substitua-as");
1636:          delay_ms(4000);
1637:          lcd_custom_cmd(lcd_clear);
1638:          lcd_custom_out(1,1,"BF");
1639:       }
1640:       return;
1641: }
1642: void bateria2()
1643: {
1644:       if (portb.f0 == 0)
1645:       {
1646:          lcd_custom_out(1,1,"BF");
1647:       }
1648:       return;
1649: }
```

Anexo B . Análise Estática dos Componentes Mecânicos

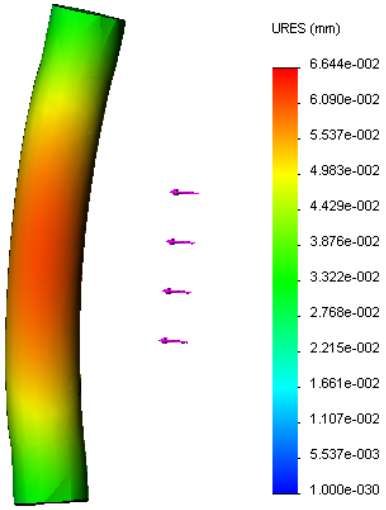


Figura 55. Distribuição das deslocações no veio do rolete com um factor de escala de 112.

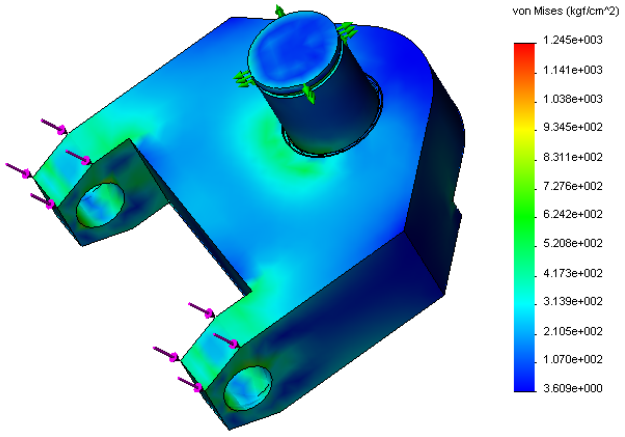


Figura 56. Distribuição das tensões no suporte do rolete e respectivo veio com um factor de escala de 1.

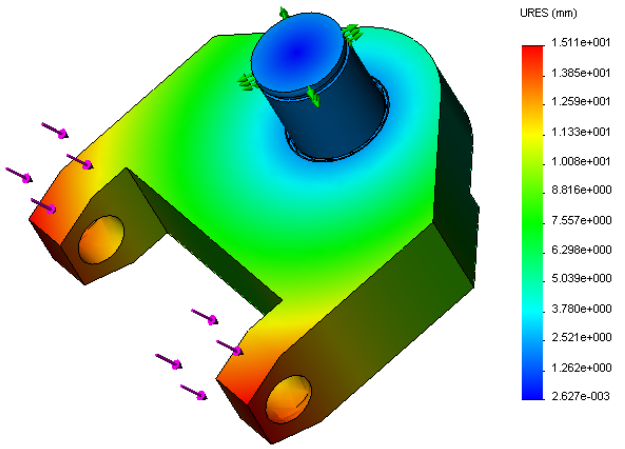


Figura 57. Distribuição dos deslocamentos no suporte do rolete e respectivo veio com um factor de escala de 1.

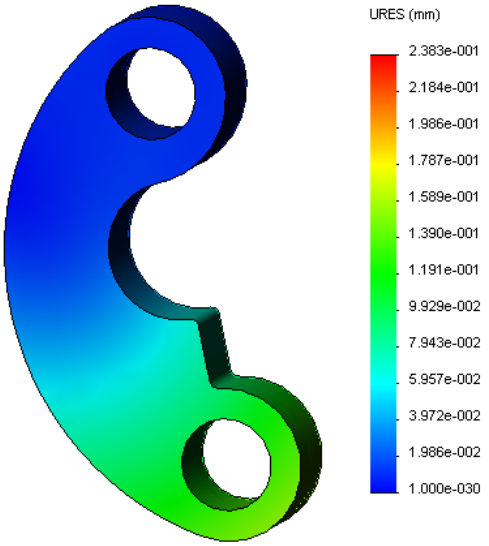


Figura 58. Distribuição dos deslocamentos na biela com um factor de escala de 1.

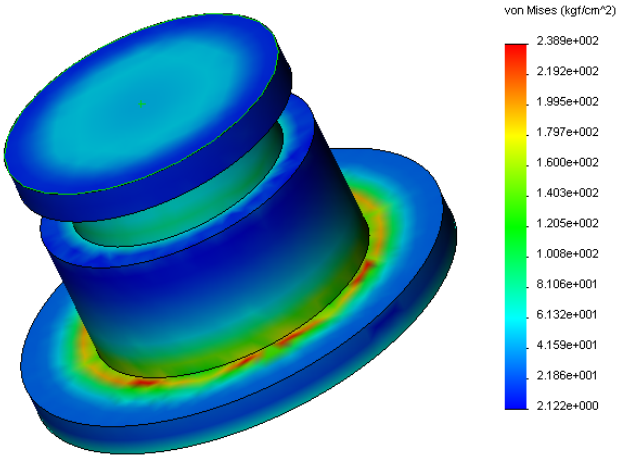


Figura 59. Distribuição das tensões na pata da cassete com um factor de escala de 1.

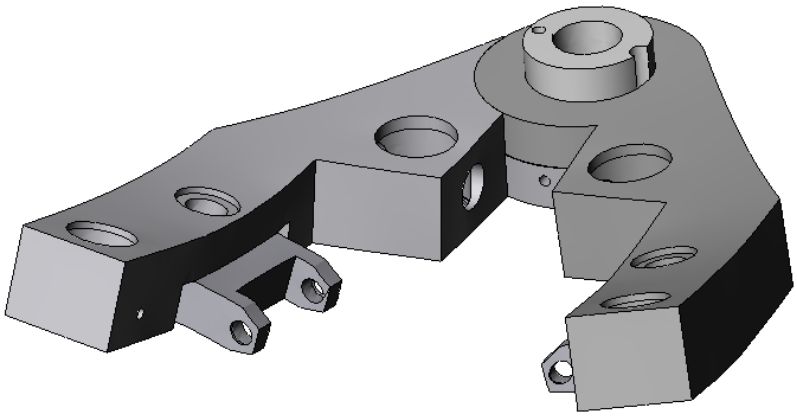


Figura 60. Deformações na tesoura com um factor de escala de 100.

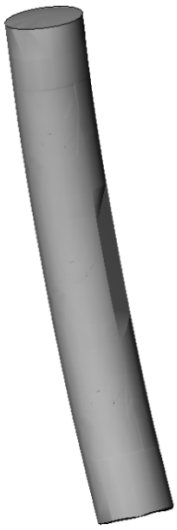


Figura 61. Deformações no veio do rolete com um factor de escala de 100.

Figura 62. Deformações na biela com um factor de escala de 100.

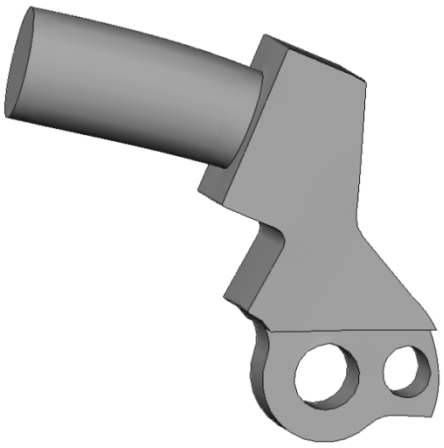


Figura 63. Deformações na manivela com um factor de escala de 50.

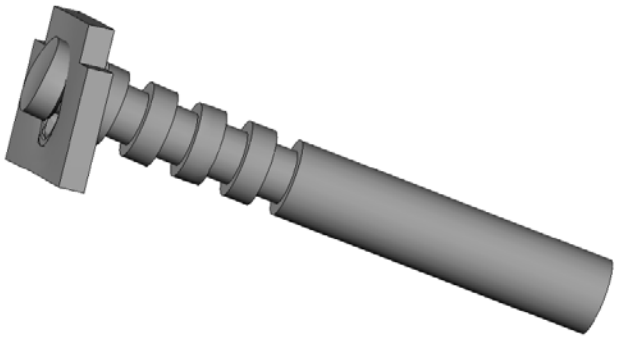


Figura 64. Deformações no conjunto guilhotina/veio escalonado com um factor de escala de 100.

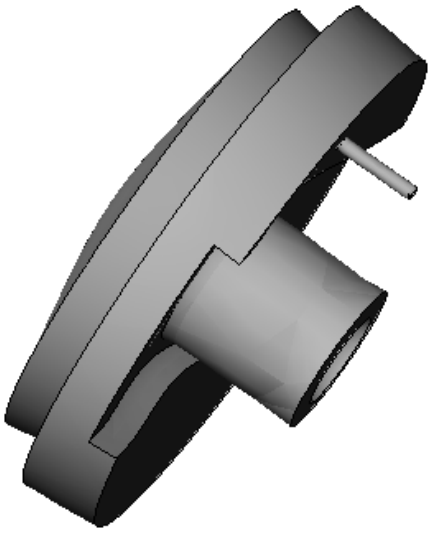


Figura 65. Deformações na cassete com um factor de escala de 100.

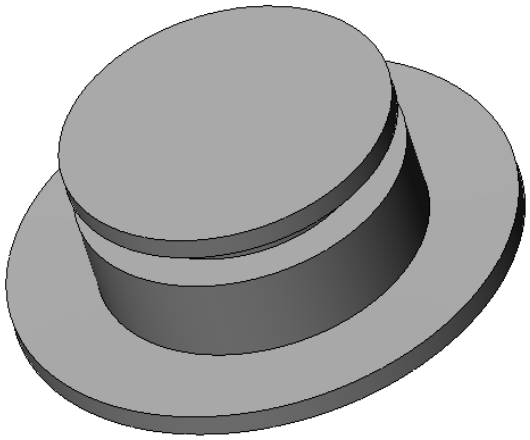


Figura 66. Deformações na pata da cassete com um factor de escala de 100.

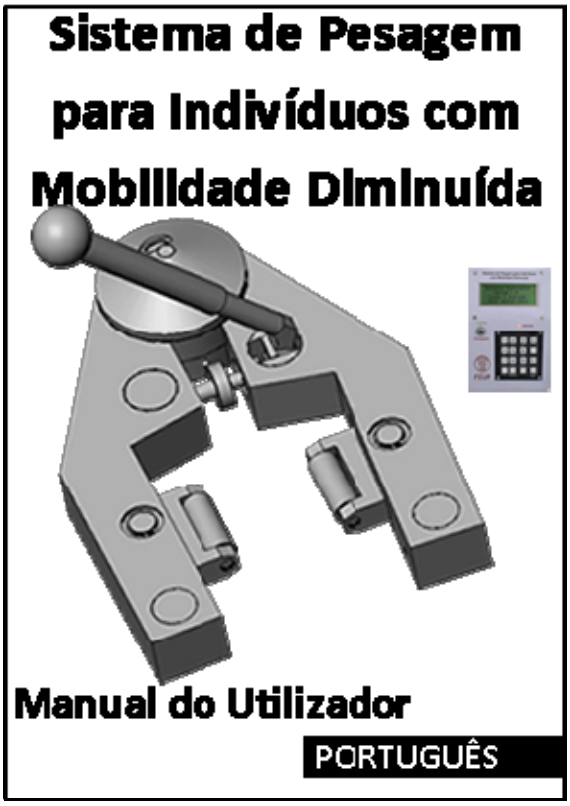
Anexo C . Orçamento Eléctrico/Electrónico do Protótipo

Tabela 10. Orçamento eléctrico/electrónico do protótipo.

Descrição dos componentes	Custo unitário	Quantidade	Custo total
PIC18LF4553	8,00 €	1	8,00 €
Flexiforce A201	20,25 €	12	242,96 €
LCD STN transflectivo 20x4	16,41 €	1	16,41 €
C091A frontal 6vias fêmea	5,39 €	4	21,56 €
C091A frontal 6vias macho	5,45 €	4	21,80 €
C091A macho roscado	8,56 €	4	34,24 €
C091A fêmea roscada	7,82 €	4	31,28 €
suporte baterias 2xAA	0,34 €	1	0,34 €
suporte baterias 4xAA	0,59 €	1	0,59 €
OP297	6,72 €	8	53,76 €
M2x16mm 100unidades	1,50 €	1	1,50 €
M2 50unidades	2,84 €	1	2,84 €
Condensador elec 100 uF 16 V	0,02 €	1	0,02 €
Terminal 40 vias para cabo IDE	3,57 €	1	3,57 €
MAX667	5,65 €	1	5,65 €
Resistência variável	1,89 €	1	1,89 €
Resistências	0,08 €	14	1,12 €
Resistências de precisão	1,05 €	20	21,00 €
Max681	9,66 €	1	9,66 €
Interruptor	4,34 €	1	4,34 €
Teclado	8,60 €	1	8,60 €
Baterias AA	4,50 €	6	27,00 €
Consola	15,52 €	1	15,52 €
Cabo plano 30 metros	78,45 €	0,0004	0,03 €
Placas de circuito impresso	10,00 €	1	10,00 €
Conector macho cabo plano	0,69 €	2	1,38 €
Construção mecânica (oficinas)	- €	1	- €
Recursos Humanos	- €	1	- €

TOTAL	545,06 €

Anexo D . Manual do Utilizador



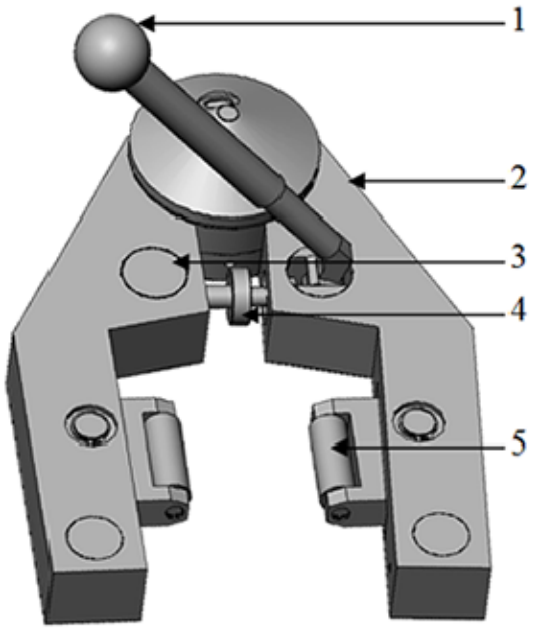
Tiago Faustino Andrade

Julho de 2008

Índice

Aplicação da Sapata.....2
Consola de Comando.....4
Funções e Valores do Teclado.....5
Menu Tarar.....7
Menu Pesagem.....17

Aplicação da Sapata



2

Legenda:

1. Alavanca
2. Tesoura
3. Botão
4. Volante
5. Rolete (x2)

A elevação de cada roda da cama deve iniciar-se pela aproximação da tesoura (1) à roda da cama. De seguida abre-se a tesoura através de um botão (3), solta um veio que liga os dois braços da tesoura, e ajustam-se os roletes à roda. Caso não se consiga fazer um bom ajuste através do botão, pode-se recorrer a um volante (4) que permite um ajuste mais fino. Tendo-se os roletes bem posicionados, tangentes à roda e paralelos entre si, exerce-se força na alavanca (1), descolando a roda do chão a uma determinada altura.

3

Consola de Comando

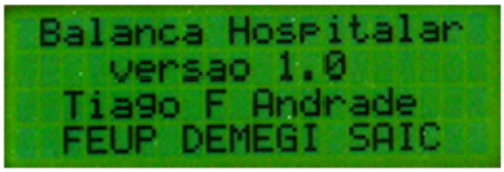
4

Funções e Valores do Teclado

TECLA	FUNÇÕES	VALORES
1	escrita e selecção de memória	1 A B C
2	escrita e selecção de memória	2 D E F
3	escrita e selecção de memória	3 G H I
A	desligar alarme e escrita	- * _ +
4	escrita e selecção de memória	4 J K L
5	escrita e selecção de memória	5 M N O
6	escrita e selecção de memória	6 P Q R
.	escrita	. , : ;
7	escrita e selecção de memória	7 S T U
8	escrita e selecção de memória	8 V W X
9	escrita e selecção de memória	9 Y Z Ç
C	cancelar	
0	escrita	0 ? ( )
←	navegar menu e escrita	apagar
↓	navegar menu e escrita	
→	navegar menu e escrita	espaço

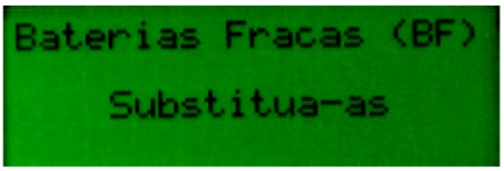
5

Apresentação



Menu principal

Informação de baterias fracas



6

Menu Tarar

No Menu Tarar pretende-se memorizar valores de taras para serem utilizadas à posteriori no Menu Pesar. Neste menu o utilizador pode optar por fazer uma tara automática, uma tara manual ou simplesmente visualizar as taras guardadas em memória, tendo a possibilidade, nesta fase, de editar os campos que contêm a informação do valor da tara e da identificação associada.

Na tara automática, o sistema apresenta o somatório dos valores lidos nos 12 elementos sensores. Dando a ordem de memorização do valor da tara, o sistema pesquisa as memórias livres na

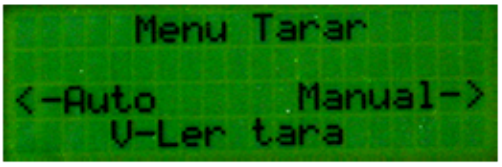
7

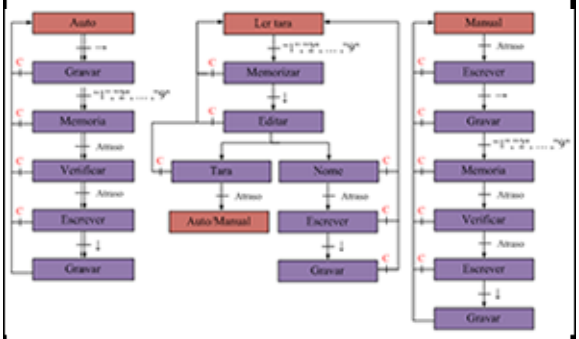
eeprom, disponibilizando a informação no LCD para que o utilizador faça a selecção desejada.

Na tara manual, o sistema permite ao utilizador digitar o valor da tara que pretende, seguindo-lhe o mesmo processo de memorização descrito anteriormente. A figura seguinte é constituída por um diagrama de blocos que pretende dar uma ideia das funções disponibilizadas no Menu Tarar.

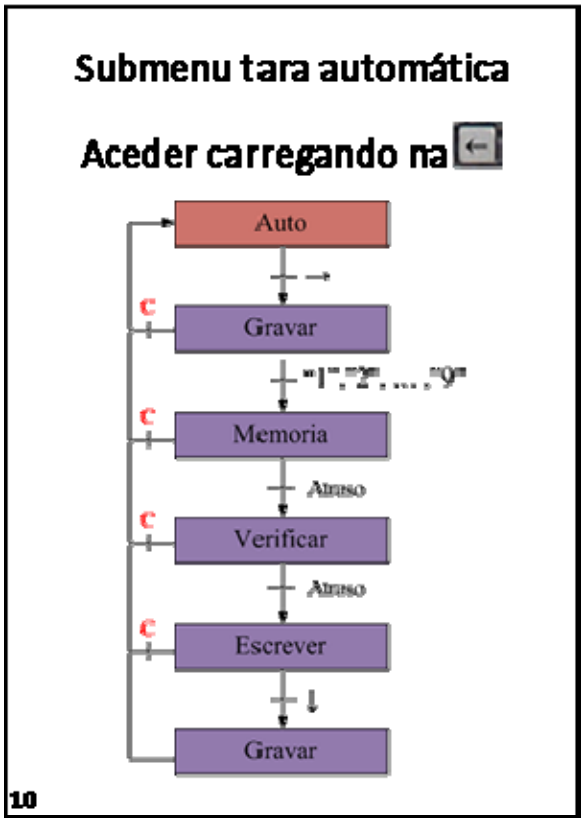
8

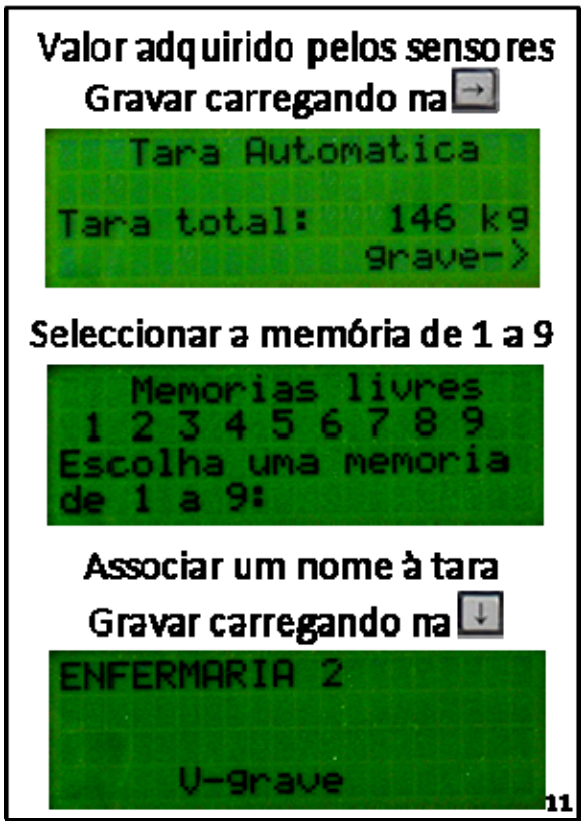
Menu Tarar

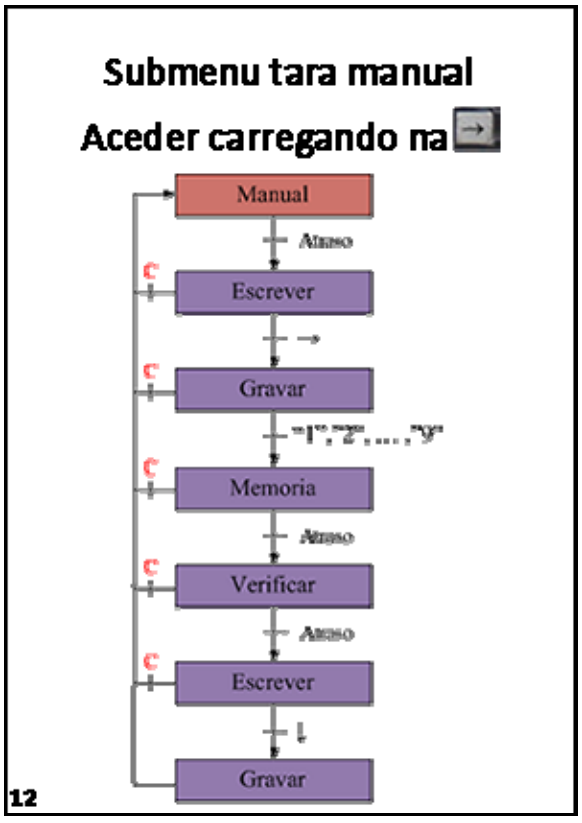


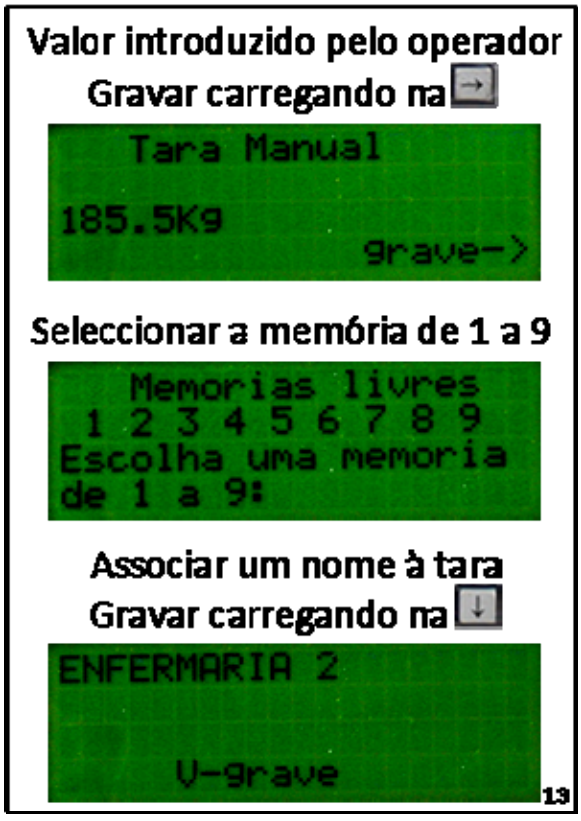


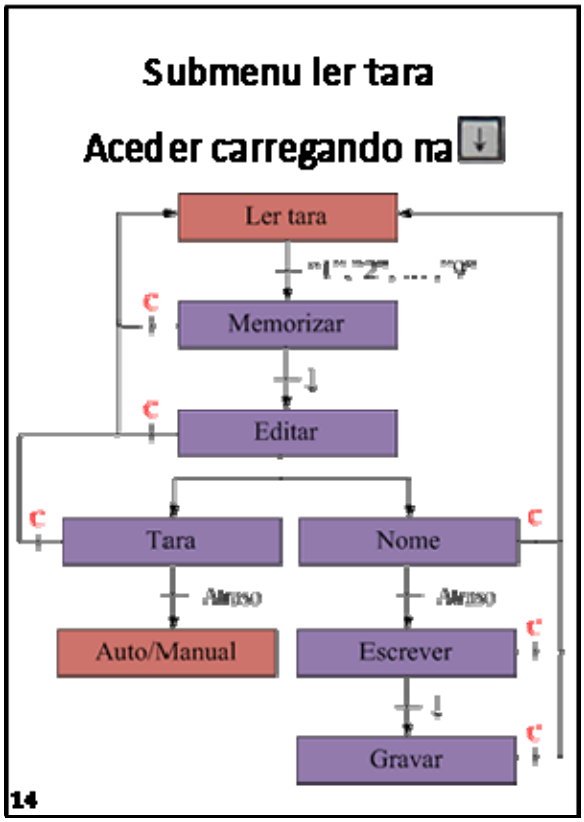
9

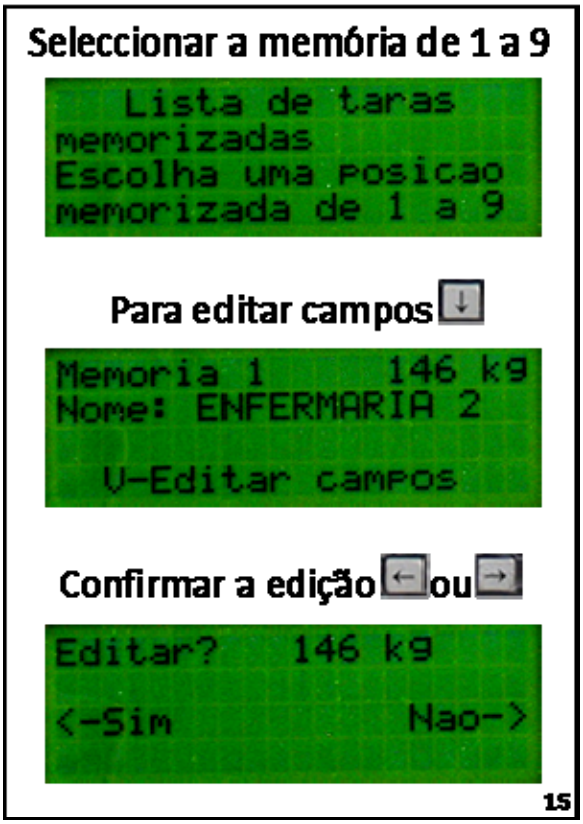

Submenu tara automática
Aceder carregando na
Auto
→
Gravar
"1","2",...,"9"
Memoria
Atraso
Verificar
Atraso
Escrever
↓
Gravar
10

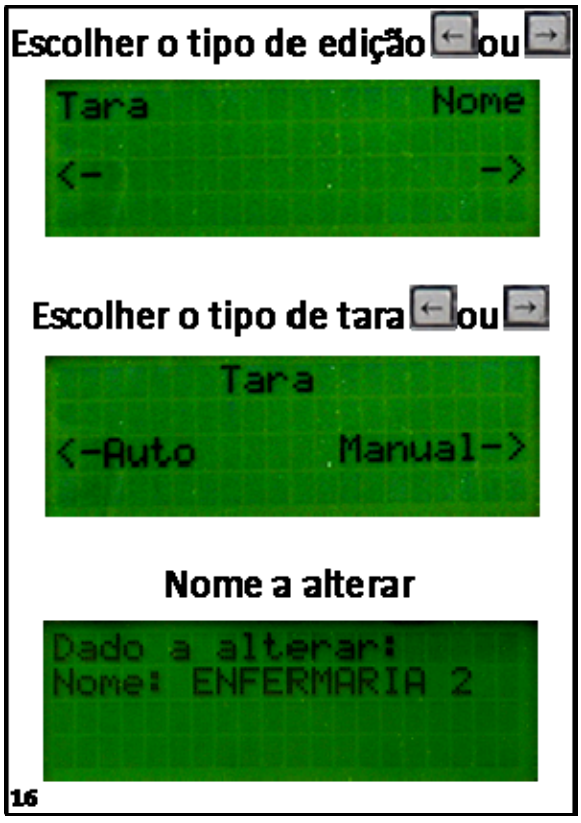

Valor adquirido pelos sensores
Gravar carregando na
Tara Automatica
Tara total: 146 kg
grave->
Seleccionar a memória de 1 a 9
Memorias livres
1 2 3 4 5 6 7 8 9
Escolha uma memoria
de 1 a 9:
Associar um nome à tara
Gravar carregando na
ENFERMARIA 2
V-grave
11


Submenu tara manual
Aceder carregando na
Manual
Atraso
Escrever
→
Gravar
"1","2",...,"9"
Memoria
Atraso
Verificar
Atraso
Escrever
↓
Gravar
12


Valor introduzido pelo operador
Gravar carregando na
Tara Manual
185.5Kg
grave->
Seleccionar a memória de 1 a 9
Memorias livres
1 2 3 4 5 6 7 8 9
Escolha uma memoria
de 1 a 9:
Associar um nome à tara
Gravar carregando na
ENFERMARIA 2
V-grave
13







Menu Pesar

O Menu Pesar subdivide-se em dois menus. O primeiro permite monitorizar o peso total, ou seja, o peso do paciente mais o peso da cama e equipamento associado e no segundo o peso do paciente, isto é, o peso total com a subtracção da tara memorizada anteriormente.

No menu referente ao peso do paciente, deve definir-se, previamente, um valor crítico para a variação do peso a monitorizar e o valor da tara anteriormente memorizada. As configurações fornecidas numa primeira pesagem, variação crítica e tara, são memorizadas para que na segunda

17

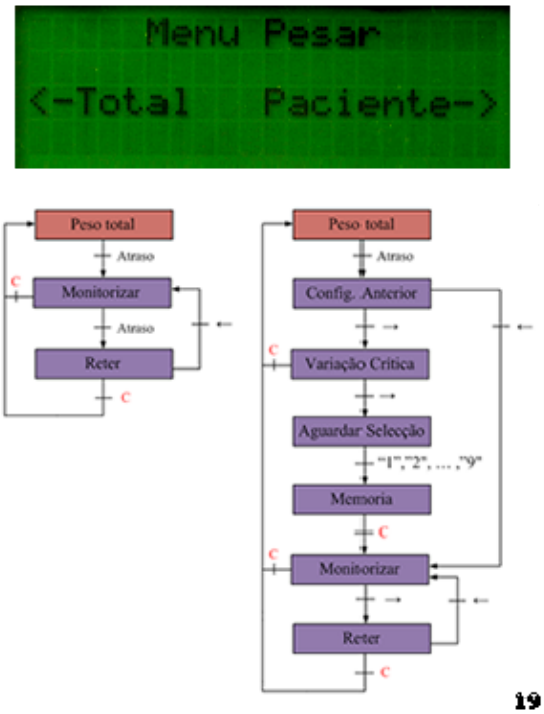
pesagem do mesmo indivíduo não seja necessário introduzir novamente os mesmos dados. Naturalmente que se a balança for utilizada num outro paciente, há a possibilidade de mudar essa configuração. A variação crítica introduzida tem como objectivo alertar o pessoal hospitalar para a alteração do peso, sendo isso feito através de um LED que pode ser apagado através da tecla A.

Durante a monitorização, seja no menu peso total ou peso do paciente, há possibilidade de reter o valor lido no LCD, retomando-se de seguida a monitorização por ordem do operador.

Na figura seguinte é mostrado um diagrama de blocos que caracteriza as diversas fases do Menu Pesar.

18

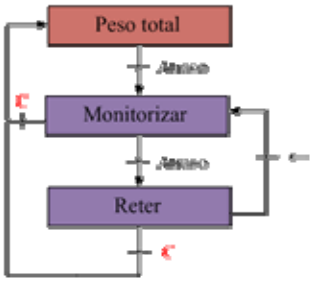
Menu Pesar



19

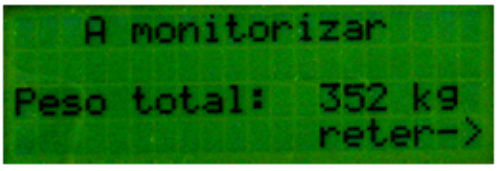
Submenu peso total

Aceder carregando na ←

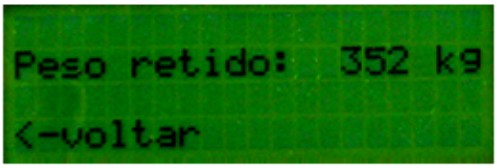


20

Reter valor carregando na →

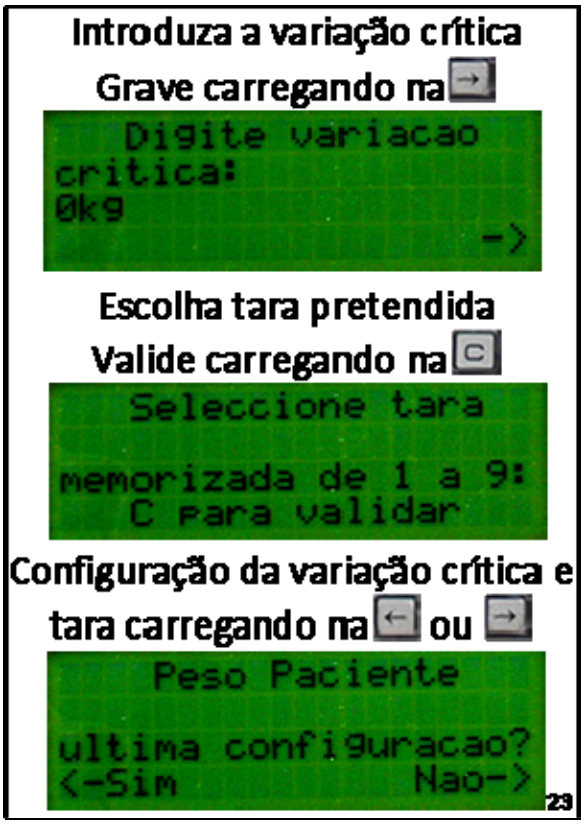


Voltar à monitorização carregando na ←



21

Submenu peso paciente
Aceder carregando na
Peso total
Config. Anterior
Variação Crítica
Aguardar Selecção
Memoria
Monitorizar
Reter
22


Introduza a variação crítica
Grave carregando na
Digite variacao
critica:
0kg
->
Escolha tara pretendida
Valide carregando na
Seleccione tara
memorizada de 1 a 9:
C para validar
Configuração da variação crítica e
tara carregando na ou
Peso Paciente
ultima configuracao?
<-Sim
Nao->
23

Universidade do Porto
Faculdade de Engenharia
FEUP
DEMEGI
SAIC
Tiago Faustino Andrade

Anexo E . Esquemas Eléctricos/Electrónicos

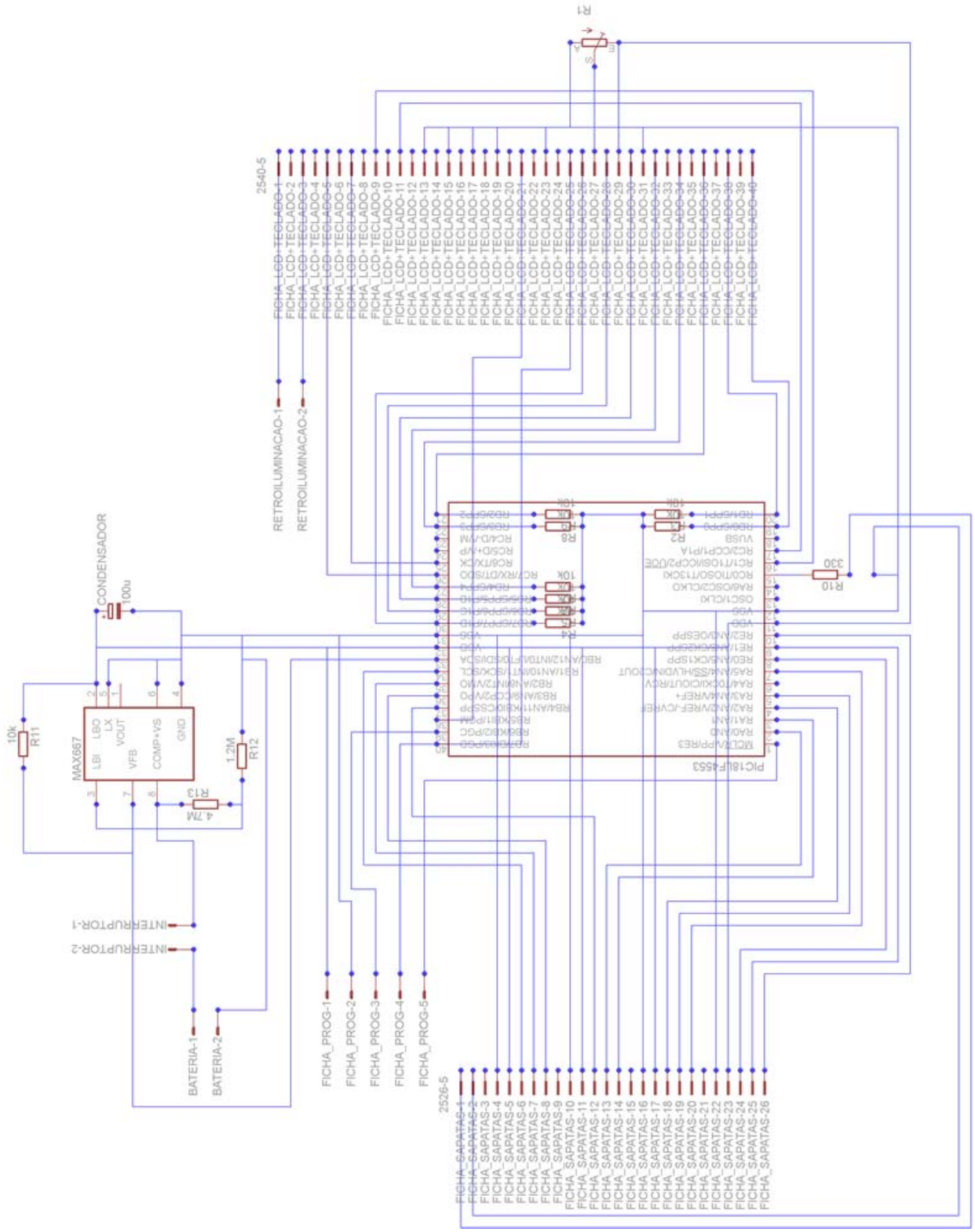


Figura 67. Circuito eléctrico da consola.

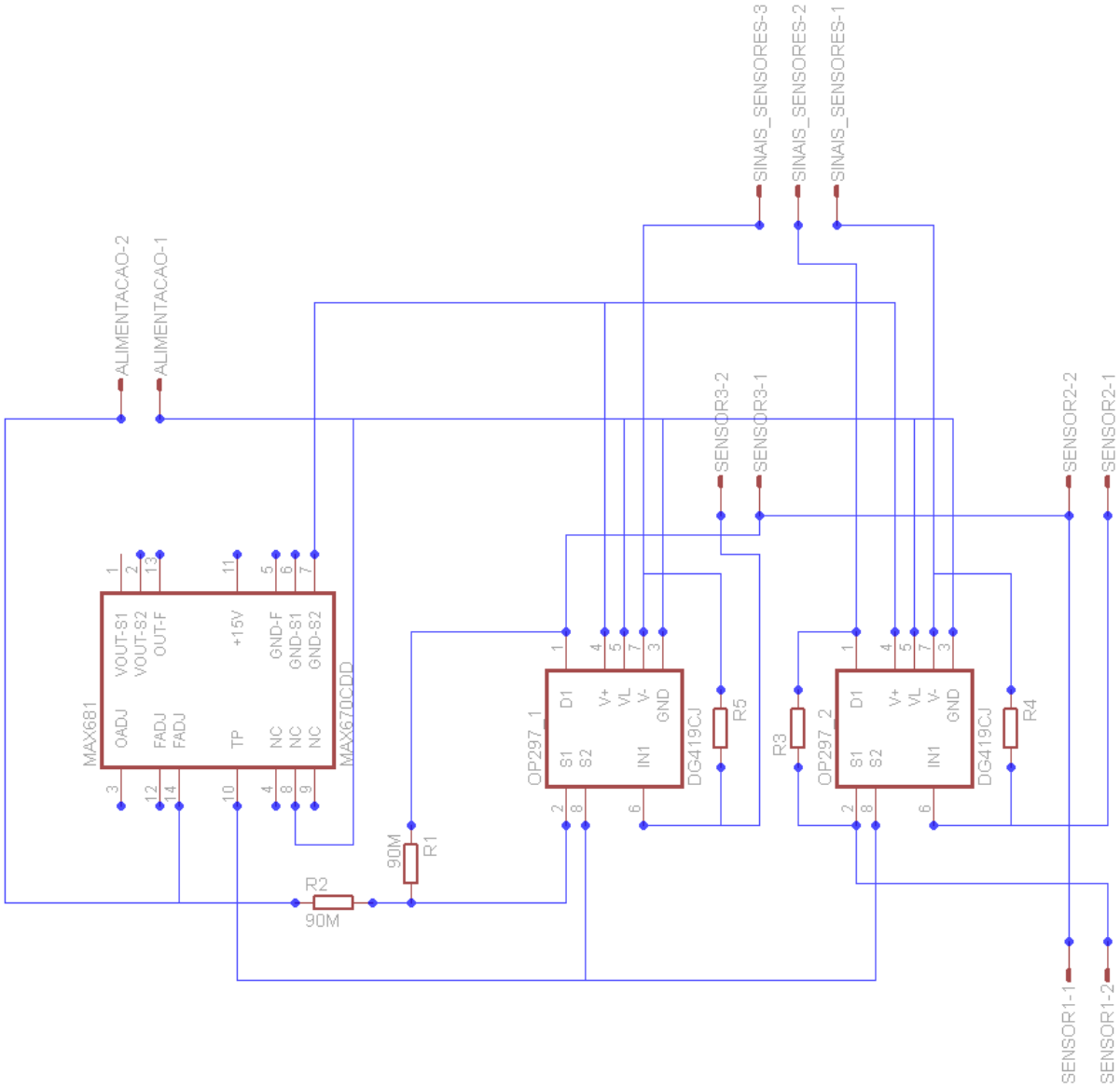


Figura 68. Circuito eléctrico da sapata.